The Thesis Committee for Christopher Matthew Bender certifies that this is the approved version of the following thesis:

Three-Dimensional Geoacoustic Perturbative Inverse Technique for the Shallow Ocean Water Column

APPROVED BY

SUPERVISING COMMITTEE:

Preston S. Wilson, Supervisor

Megan S. Ballard, Supervisor

Three-Dimensional Geoacoustic Perturbative Inverse Technique for the Shallow Ocean Water Column

by

Christopher Matthew Bender, B.S.M.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2012

To my family

Sarah, my heart and soul

Carl and Melinda, my foundation

Michael and Ashley, my partners in crime

Acknowledgments

I would like to acknowledge Dr. Ballard and Dr. Wilson for helping me along this part of my journey from ignorance to understanding. I have been very appreciative of their assistance and insight as we explored this part of the ocean. I am especially grateful to Dr. Ballard for the time and effort she put in to help me succeed. Her encouragement and confidence were essential in making this research an enjoyable, educational process. I would also like to thank Dr. Becker and Dr. Frisk for allowing me to participate in the MOMAX V experiment, Bryant Tran for his assistance in furthering my understanding of coding, and Dr. Metzler for his advice on presentation.

Christopher Matthew Bender

Three-Dimensional Geoacoustic Perturbative Inverse Technique for the Shallow Ocean Water Column

Christopher Matthew Bender, M.S.E.
The University of Texas at Austin, 2012

Supervisors: Preston S. Wilson
Megan S. Ballard

This work focuses on developing an inversion scheme to estimate water-column sound-speed fields in three dimensions. The inversion scheme is based on a linearized perturbative technique which utilizes estimates of modal travel times. The technique is appropriate in the littoral ocean where measurements are made across range and cross-range distances greater than 10 km to ensure sufficient modal dispersion. Previous applications of then inversion technique has been limited to one or two dimensions and/or focused primarily on the seabed.

Compared to past applications, the accuracy and uncertainty of the solution is improved by employing approximate equality constraints within the context of *a priori* estimates of model and data covariances. The effectiveness of the constrained technique is explored through a one-dimensional example. The robustness of the technique is illustrated by introducing different types

of errors into the inversion and considering the accuracy. A further examination of the technique is given by exploring a three-dimensional example. Several case studies are presented to investigate the effects of different levels of environmental variability and spatial sampling.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The ocean covers approximately 71% of the Earth's surface. It is an essential part of our ecosystem, economy, and military. It is a commonly held belief amongst scientists that life started in the ocean. It is impossible to deny the extraordinary reliance life on our planet has on its oceans. Without the ocean, life on Earth could not exist (at least not to the extent and variation that it does). For these reasons, it is necessary and good to understand and explore the ocean both for the past, present, and future, for protection and cooperation, and for enhancing our understanding of the universe and ourselves.

1.1 Ocean Acoustics

The premier tool for remote sensing in the ocean is sound. This is because the absorption of humanities' normal mode of sensing, electromagnetic energy, is far too high in water to be useful at significant range [1]. Acoustic waves (sound waves) have a considerably lower absorption in water and can be used to sense at scales ranging from micrometers to kilometers.

Acoustics has many applications in the ocean. It is used to image the

ocean floor, to search for targets in the water, and for communication with underwater vehicles. However, all of these applications require some knowledge of the acoustic properties of the water, most importantly sound speed. Knowledge of the sound speed can be used to gain information about other environmental conditions, such as temperature. It is possible to make these inferences because sound speed has a high correlation with temperature. This allows for approximation of one property from the other. Other parameters known to effect sound speed include salinity and pressure.

When knowledge of the sound speed profile is only needed locally, it is usually sufficient to take a direct measurement of conductivity, temperature, and depth (CTD) using on-board or moored equipment and applying those measurements to estimate sound speed. However, when the region of interest is large, it becomes infeasible to make sufficiently dense point measurements to adequately characterize the region. Fortunately, it is possible to make volumetric estimates through the use of inverse analysis.

1.2 Approach

This work is concerned with determining the sound speed in a three-dimensional (3D) volume of water in the shallow ocean through the application of inverse methods. The technique explored in this thesis was developed by Rajan in 1987 [2] and utilizes a perturbative method based on differences in modal travel times caused by dispersion to determine sound speeds in the water column in 3D.

This technique addresses common issues of stability and uniqueness present in inversion problems through the use of approximate equality constraints [3]. This technique allows for both the inclusion of *a priori* knowledge of the region and/or uncertainties in the sea floor. This is made possible through the use of an assumed starting profile about which the perturbation is based and by controlling which modes are used in the inversion process. By including only low-order modes, it is possible to gain good results for water-column inversions with virtually no information about the sea floor [4].

1.3 Background

There are two primary methods used to solve the nonlinear geoacoustic inversion problem: Monte Carlo and locally-linear techniques. Monte Carlo search methods employ matched-field inversions and have been widely applied to estimate properties of the seafloor [5–7]. These methods characterize the ocean as several layers over a half space. Each layer is uniquely defined based on several environmental parameters. The specific parameters vary between techniques but can include, longitudinal sound speed, transverse sound speed, density, and attenuation. These Monte Carlo methods are most effective in range-independent environments. When range-dependence is introduced the parameter search space and computational workload increases exponentially.

Locally-linear techniques apply linear approximations to the nonlinear problem to take advantage of real-world measurements to drive the inversion. Several locally-linear techniques exist to invert for the ocean sound speed in

3D [8–10]. These techniques take advantage of measurements of acoustic time of flight to drive the inversion. These methods assume the waves travel along rays which is inappropriate for the shallow ocean.

This work is based on a locally-linear technique developed by Rajan [2] for estimating sound speed in the shallow ocean sediment. Rajan's techniques allow for two related methods of estimating sound speed in one dimension (1D). The first method exploits estimates of modal wave numbers and has been used with great success by a number of investigators [4, 11–14]. This method has been expanded upon to perform inversion in the water column [15].

The second method utilizes estimates of modal travel time along a path of known length. By using a number of paths that exist in range and cross-range this method can be used to invert for the sound-speed field in 3D. This technique has been used to estimate sediment sound speeds in range-dependent environments [11]. This work will utilize modal travel time estimates to invert for the sound-speed field in 3D in the water column only. Employment of additional environmental knowledge and mathematical techniques allows for inversion of the sound-speed field in both the water column and the sediment, similar to what has been done in 1D [16].

1.4 Thesis Roadmap

Chapter 2 explores the physical phenomena known as acoustics. A brief explanation of general acoustics is made and the particular form of analysis

termed normal mode theory is discussed. The explanation is focused down to bounded environments known as waveguides. Several of the effects of waveguides are discussed with a primary focus on modal dispersion.

Chapter 3 introduces the derivation of the 3D inversion scheme which will be employed. Approximate equality constraints are offered as a method for controlling the inherent instability of the inverse problem. A discussion of resolution and error analysis is introduced as a measure of how well the inversion can be expected to and will perform.

Chapter 4 presents the results of the inversion on simulated data. The use of simulated data allows for comparisons to known truth. This allows for an illustrative method to explore the strengths and limitations of the inversion technique. Examples are generated using both acoustic simulation and oceanographic simulation. A case study is performed on the effects of differing levels of spatial sampling and variability.

Chapter 5 summarizes the work in the previous chapters and remarks on potential future works.

Chapter 2

Background

2.1 Introduction

The most common problem in acoustics is determining the pressure field created by an acoustic source in a known environment. There are five common methods to model the acoustic propagation in the ocean: normal modes, method of images, rays, the parabolic equation, and finite difference or finite element [17]. The primary focus of this work is the method of normal modes. This chapter will attempt to provide the necessary background understanding of normal modes to provide context for subsequent discussion. The chapter begins with a brief explanation of the appropriateness of utilizing normal modes in the shallow ocean. Finally, a discussion of group speed is made within the context of cylindrical waveguides.

2.2 Normal Modes

The best way to determine which solution method is most effective is to compare the number of unique identifiers that are necessary to model the pressure field. For normal mode analysis this is the number of modes; for ray theory, the number of rays; for finite differences, the number of grid points; etc.

The number of identifiers for each method can be determined prior to solving the problem. This is done based upon the length scale of the environment and the length scale of the acoustics (the wavelength).

Generally, the necessary number of elements or grid points is significantly greater than the number of modes or rays. So, the most common comparison in determining methods is between number of modes and the number of rays. A comparison of the acoustic and environmental length scales results in the following relationship [18]

$$R\lambda = 2D^2 \tag{2.1}$$

where R is range, D is the average water depth, and λ is the acoustic wavelength. At $R\lambda < 2D^2$ the number of rays is less than the number of modes and ray theory is more convenient. At $R\lambda > 2D^2$ the number of rays is greater than the number of modes and normal mode theory is more convenient. The focus of this work requires the use of normal modes.

The transition between inequalities derived from Eq. (2.1) is a good indicator of where the inversion will be most effective or what frequencies should be used for a given environment. The inversion is not limited by the equation, but the complexity of the problem increases as $\frac{R\lambda}{2D^2}$ tends to zero. The increased complexity is problematic as it becomes more difficult to distinguish between modal arrivals.

2.3 Waveguides

The simplest model of the ocean consists of two parallel reflecting planes separated by a fluid medium. This bounded region conforms to the definition of a waveguide, where the bound dimension is depth and the propagating dimension is range. The appropriate governing equation is referred to as the inhomogeneous Helmholtz equation

$$\rho(z)\,\frac{\partial}{\partial z}\left\{\frac{1}{\rho(z)}\frac{\partial p_\omega(r,z)}{\partial z}\right\}+\frac{\partial^2 p_\omega(r,z)}{\partial r^2} + \frac{1}{r}\frac{\partial p_\omega(r,z)}{\partial r}+k_0^2\,(z)\,p_\omega(r,z)=-F(r,z) \qquad (2.2)$$

where $\rho(z)$ is the density as a function of depth only, $k_0 = \dfrac{\omega}{c_0(z)}$ is the total acoustic wave number, $F(z)$ is some source function centered at zero range, and $p_\omega(r,z)$ is the acoustic pressure with subscript ω to clarify an assumed time dependence has been made; namely, $p(r,z,t) = p_\omega(r,z)\,e^{i\omega t}$. This equation assumes axial symmetry.

For the time being, it is convenient to assume that density and sound speed are constants such that Eq. (2.2) reduces to the more familiar Helmholtz equation

$$\frac{\partial^2 p_\omega}{\partial z^2}+\frac{\partial^2 p_\omega}{\partial r^2}+\frac{1}{r}\frac{\partial p_\omega}{\partial r}+k_0^2 p_\omega=-F(r,z) \qquad (2.3)$$

where the arguments of p_ω are unchanged but have been removed for concise-

ness. This equation is well understood and many solution techniques exist, including, separation of variables [19], Green's functions [20], and spectral integration [21].

2.3.1 The Ideal Waveguide

As mentioned previously, the simplest model of the ocean consists of two parallel planes separated by a fluid medium. The Helmholtz equation accounts for the fluid medium, but it is still necessary to ascribe boundary conditions to the two "plates." In the ocean, this is best done by requiring a pressure release condition at the ocean's surface and a rigid boundary at the ocean's floor. Waveguides with these particular boundary conditions will be referred to as "ideal waveguides." These two conditions are expressed mathematically as

$$p_\omega(r, 0) = 0 \tag{2.4a}$$

$$\frac{\partial p_\omega(r, D)}{\partial z} = 0. \tag{2.4b}$$

The solution to Eq. (2.3) under the conditions in Eq. (2.4) with no forcing function ($F(r, z) = 0$) result in a solution of the form

$$p_\omega(r, z) = \sum_{n=1}^{\infty} A_n \sin(k_{zn} z)\, H_0^{(2)}(k_{rn} r) \tag{2.5}$$

where A_n is a modal source-dependent amplitude term, k_{zn} is the vertical

wavenumber, k_{rn} is the horizontal wavenumber, and $H_0^{(2)}$ is taken as the zero order outgoing Hankel function. Additionally, the values of k_{zn} and k_{rn} are given as

$$k_{zn} = \frac{2n-1}{2}\frac{\pi}{D} \quad , \quad n = 1, 2, 3, ... \tag{2.6a}$$

$$\begin{aligned} k_{rn} &= \sqrt{k_0^2 - k_{zn}^2} \\ &= \sqrt{k_0^2 - \left(\frac{2n-1}{2}\frac{\pi}{D}\right)^2} \\ &= k_0\sqrt{1 - \left(\frac{2n-1}{4}\frac{c_0}{fD}\right)^2}. \end{aligned} \tag{2.6b}$$

If a simple source is placed at depth z_s the value of A_n reduces to [17]

$$A_n = -\frac{iS_w}{2D}\sin(k_{zn}z_s)\,. \tag{2.7}$$

where S_w is the source strength.

When the horizontal wave number is imaginary, the wave no longer propagates. Such "waves" are referred to as evanescent waves and decay exponentially. Equation (2.6b) indicates that the number of propagating modes in the waveguide is restricted and dependent on frequency in a nonlinear manner. This frequency dependence of the normal modes is known as dispersion and will be the key to the inversion method.

The frequency at which the wave becomes evanescent is known as the cut off frequency and is dependent on mode number. A simple expression can be found for the cut off frequency by setting the argument of the radical to zero and solving for frequency:

$$1 - \left(\frac{2n-1}{4} \frac{c_0}{f_{n,c} D} \right)^2 = 0$$

$$f_{n,c} = \frac{2n-1}{4} \frac{c_0}{D} \tag{2.8}$$

where $f_{n,c}$ is the cut off frequency of the nth mode.

2.3.2 Phase Speed

The phase velocity is the speed at which points of constant phase travel down the waveguide. To determine the phase speed, it is necessary to express the phase as a function of time and range. The phase can be expressed simply and without approximation as

$$\begin{aligned} \Phi &= \arg(p(r,z,t)) \\ &= \omega t - \arctan\left(\frac{N_0(k_{rn} r)}{J_0(k_{rn} r)} \right) - \frac{\pi}{2} \end{aligned} \tag{2.9}$$

where Φ is the phase, J_0 is the ordinary Bessel function of the first kind of order zero, and N_0 is the ordinary Bessel function of the second kind (also known as the Neumann or Weber function) of order zero. Equation (2.9) is exact for

$r > 0$. If r were allowed to be negative, the Neumann function would not be purely real and the analysis would be considerably more complex. Fortunately, in cylindrical coordinates, r is always nonnegative.

If Φ is differentiated and set to zero

$$d\Phi = \omega dt - 2k_{rn}\left(\pi k_{rn} r\left(J_0^2(k_{rn}r) + N_0^2(k_{rn}r)\right)\right)^{-1} dr = 0 \tag{2.10}$$

an exact expression for the phase speed can be found by solving for $\dfrac{dr}{dt}$:

$$\frac{dr}{dt} = \frac{\omega}{k_{rn}}\frac{\pi}{2}k_{rn} r\left(J_0^2(k_{rn}r) + N_0^2(k_{rn}r)\right). \tag{2.11}$$

While this expression is exact, it is complicated and not intuitive. Therefore, to aid in simplicity and understandability, it is helpful to expand the bracketed terms into a truncated Laurent series of two terms. Maintaining two terms allows for a first-order approximation and an estimate of the next order error. The use of only two terms will be explained shortly. A Laurent series is preferred over a power series due to the ability to handle the singularity [22] in the Neumann function and so that additional terms will decrease in magnitude for large values of $k_{rn}r$, which is typical of ocean acoustics and necessary for this work.

After some manipulation, the desired result can be shown as:

$$\begin{aligned}\frac{dr}{dt} &= \frac{\omega}{k_{rn}}\frac{\pi}{2}k_{rn}r\left(\frac{2}{\pi}\frac{1}{k_{rn}r}-\frac{7}{64\pi}\left(\frac{1}{k_{rn}r}\right)^3+O\left([k_{rn}r]^{-5}\right)\right)\\ &\simeq \frac{\omega}{k_{rn}}\left(1-\frac{7}{128}\left(\frac{1}{k_{rn}r}\right)^2\right).\end{aligned} \tag{2.12}$$

This result shows that for $k_{rn}r \gg 1$ the second order correction can be neglected and the phase speed can be written as

$$c_{\mathrm{ph},n} = \frac{\omega}{k_{rn}} \tag{2.13a}$$

$$c_{\mathrm{ph},n} = \frac{c_0}{\sqrt{1-\left(\frac{2n-1}{4}\frac{c_0}{fD}\right)^2}} \quad , \text{ ideal waveguide} \tag{2.13b}$$

which is the same as for a two-dimensional (2D) waveguide in Cartesian coordinates. Figure 2.1 illustrates the differences between the exact expression, a first-order approximation, and a second order approximation as a function of normalized range $k_{rn}r$. Normalized phase speed is the phase speed at each order divided by the phase speed at infinite range. This result is completely expected

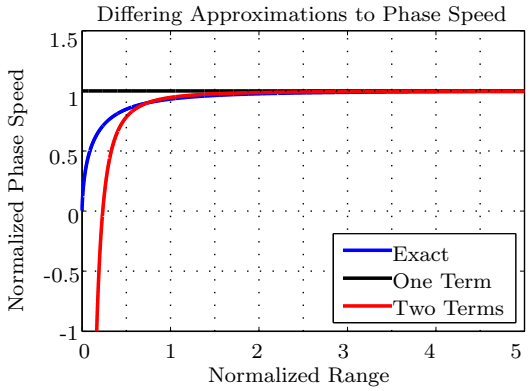


Figure 2.1: Comparison of the accuracy of different phase speed approximations

since for $r \gg \lambda_{rm}$ cylindrical waves can be approximated as plane waves, where $\lambda_{rm} = {}^{2\pi}/_{k_{rm}}$.

Figure 2.2 shows the phase speed as a function of frequency for the first five modes for a water-column sound speed of 1500 m/s and a water depth of 75 m. The phase speed tends to the water-column sound speed for increasing frequency (with lower order modes approaching the limit earlier) and tends to infinity as the frequency approaches cut off.

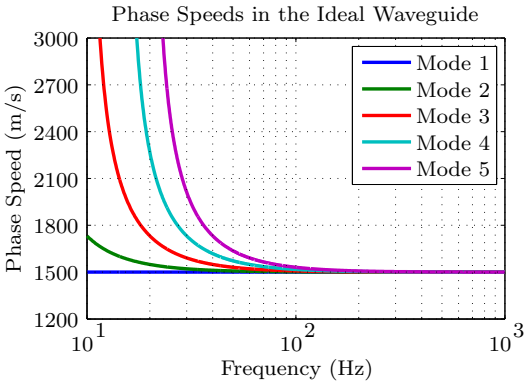


Figure 2.2: Phase speed for the first five modes of the ideal waveguide

Another interpretation of phase speed can be found by examining the wavelength trace [23]

$$\lambda_{\text{tr}} = \frac{\lambda}{\sin(\theta_{i,n})} \tag{2.14a}$$

$$\begin{aligned} c_{\text{ph},n} &= f\lambda_{\text{tr}} \\ &= \frac{c_0}{\sin(\theta_{i,n})} \end{aligned} \tag{2.14b}$$

where $\theta_{i,n}$ is the incidence angle of mode n. The incidence angle is the angle at which the mode hits the ocean bottom measured from the depth axis.

2.3.3 Group Speed

Another important effect of dispersion, one critical to the inversion technique of this paper, is the group speed. Considerable information about the shallow ocean waveguide can be extracted from group speed dispersion curves. Group speed can be thought of as the velocity of the "center of mass" of a wave packet [20]. To determine group speed it is necessary to consider a wave packet of the form $A(\omega)\, e^{i(\omega t - k_{rn}(\omega) r)}$ where $A(\omega)$ is the amplitude distribution of the wave with respect to frequency and $k_{rn}(\omega)$ is the dispersion relation. The form of this wave corresponds to a plane wave. This is justified by maintaining the limits discussed in the previous section.

The pressure field can be determined by integrating over frequency:

$$p(r,t) = \int_{-\infty}^{\infty} A(\omega)\, e^{i(\omega t - k_{rn}(\omega) r)} d\omega. \tag{2.15}$$

Expanding the dispersion relation about the transmitted frequency ω_0 to first-order ($k_{rn}(\omega) \approx k_{rn}(\omega_0) + (\omega - \omega_0)\, k'_{rn}(\omega_0)$) results in

$$p(r,t) \approx \int_{-\infty}^{\infty} A(\omega)\, e^{i\left(\omega t - k_{rn0} r - (\omega - \omega_0) k'_{rn0} r\right)} d\omega \tag{2.16}$$

where k_{rn0} and k'_{rn0} is shorthand for $k_{rn}(\omega_0)$ and $k'_{rn}(\omega_0)$, respectively. Rearranging the integral such that the terms independent of ω appear outside the integrand gives

$$p(r,t) \approx e^{-i\left(k_{rn0}-\omega_0 k'_{rn0}\right)r} \int_{-\infty}^{\infty} A(\omega)\, e^{i\omega\left(t-k'_{rn0}r\right)} d\omega. \tag{2.17}$$

Substituting $\tau = t - k'_{rn0}r$ into the integral gives

$$\begin{aligned} p(r,t) &= e^{-i\left(k_{rn0}-\omega_0 k'_{rn0}\right)r} \int_{-\infty}^{\infty} A(\omega)\, e^{i\omega\tau} d\omega \\ &= e^{-i\left(k_{rn0}-\omega_0 k'_{rn0}\right)r} p(0,\tau) \end{aligned} \tag{2.18}$$

and substituting back and taking the magnitude gives

$$|p(r,t)| = |p(0, t - k'_{rn0}r)|\,. \tag{2.19}$$

Finally, by exploiting the basic ideas of wave transport, a relationship for the rate of the energy transfer can be expressed as

$$c_{\mathrm{grp},n} = \left(\frac{\partial k_{rn}}{\partial \omega}\right)^{-1}\Bigg|_{\omega=\omega_0} \tag{2.20a}$$

$$\begin{aligned} c_{\mathrm{grp},n} &= c_0\sqrt{1-\left(\frac{2n-1}{4}\frac{c_0}{fD}\right)^2}\quad, \quad \text{ideal waveguide} \\ &= c_0 \sin(\theta_{i,n}) \end{aligned} \tag{2.20b}$$

where previous definitions of k_{rn} for the ideal waveguide have been substituted.

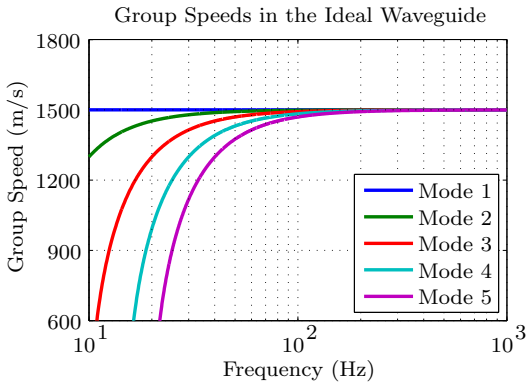


Figure 2.3: Group speed for the first five modes of the ideal waveguide

Figure 2.3 shows the group speed as a function of frequency for the first five modes for a water-column sound speed of 1500 m/s and a water depth of 75 m. The group speed tends to the water-column sound speed for increasing frequency (with lower order modes approaching the limit earlier) and tends to zero as the frequency approaches cut off. The group speed is always less than the water-column sound speed. Since the group speed is the speed at which energy is transported down the waveguide the absolute maximum can be no greater than the water-column sound speed.

This analysis also begs the question of what happens when a linear approximation is inappropriate. If there is a large enough spread in in the frequencies, the wave packet envelope, or group, changes its shape too rapidly. As a result it is impossible to observe where the "center of mass" is or how fast it moves. Morse and Ingard analyze a specific amplitude distribution (a Gaussian distribution) [20] and observe the appropriateness of a linear approximation in the familiar terms of mean frequency (ω_0) and standard deviation. In general, the more peaked $A(\omega)$ is, the more accurate the analysis. The analysis so far has assumed the amplitude distribution is a delta function, or at most, a summation of delta functions.

2.4 Summary

This chapter introduced the environment the inversion will be performed it. A brief discussion on the necessity of modal analysis in the shallow ocean was made. The approximation of the shallow ocean as an ideal waveguide was presented. Considerable effort was made in discussing the cause and effects of dispersion. Most notably, the ideas of phase and group speeds were introduced in some detail.

Chapter 3

Derivation of the Inversion Technique

3.1 Introduction

This chapter presents the technique through which measurements of the modal travel times are used to approximate the water-column sound speed in range, cross-range, and depth. The chapter begins with an introduction to the inversion algorithm. The initial solution results in an ill-posed problem in which difficulties with stability and uniqueness must be resolved. Two previous methods are presented to address these issues before a hybrid technique is derived and implemented. The corresponding method for approximating the inversion error is discussed.

3.2 The Inversion Technique

As indicated in Chapter 1, the inverse technique utilized in this paper is attributed to Rajan *et al.* [2]. The technique is perturbative, which means that the inversion output is a small correction to an assumed background profile. The technique is iterative such that the corrected profile becomes the new background profile and the inversion reapplied. Iteration continues until either the mismatch between measured data and calculated data from the

model is reduced to within user-specified tolerances, the average correction on an iteration is negligible, or a predefined number of iterations is reached.

3.2.1 Depth Separated Wave Equation

Derivation of the inverse technique originates from the depth separated normal mode equation:

$$\frac{1}{\rho(z)}\frac{d^2 Z_n}{dz^2} + \frac{d}{dz}\left(\frac{1}{\rho(z)}\right)\frac{dZ_n}{dz} + \frac{1}{\rho(z)}\left(k^2(z) - k_{rn}^2\right) Z_n = 0 \qquad (3.1)$$

where n is the mode number, Z_n is the modal eigenfunction, k_{rn} is the horizontal wave number, $\rho(z)$ is the density as a function of depth, and $k(z) = \left(\frac{\omega}{c_0(z)}\right)$ is the wave number.

A perturbation is made to the sound speed $c_0 \to c_0 + \Delta c$ which causes perturbations in the other terms: $Z_n \to Z_n + \Delta Z_n$ and $k_{rn} \to k_{rn} + \Delta k_{rn}$. Substituting into the depth separated normal modes equation yields:

$$\begin{aligned}\frac{1}{\rho(z)}\frac{d^2}{dz^2}\left(Z_n + \Delta Z_n\right) + \frac{d}{dz}\left(\frac{1}{\rho(z)}\right)\frac{d}{dz}\left(Z_n + \Delta Z_n\right) \\ + \frac{1}{\rho(z)}\left(\left(k + \Delta k\right)^2 - \left(k_{rn} + \Delta k_{rn}\right)^2\right)\left(Z_n + \Delta Z_n\right) = 0, \qquad (3.2)\end{aligned}$$

expanding to first order and removing the unperturbed terms (which reduce to zero through the unperturbed equation) gives

$$\frac{1}{\rho(z)}\frac{d^2\Delta Z_n}{dz^2} + \frac{d}{dz}\left(\frac{1}{\rho(z)}\right)\frac{d\Delta Z_n}{dz} + \frac{1}{\rho(z)}\left(k^2(z) - k_{rn}^2\right)\Delta Z_n + \frac{2}{\rho(z)}\left(k\Delta k - k_{rn}\Delta k_{rn}\right)Z_n = 0. \quad (3.3)$$

Since the unperturbed modal eigenfunctions form a complete set, any function of z can be expanded about them. Specifically,

$$\Delta Z_n(z) = \sum_m a_{nm} Z_m. \quad (3.4)$$

Substituting this expansion into Eq. (3.4) results in

$$\sum_m a_{nm}\left(\frac{1}{\rho}\frac{d^2 Z_m}{dz^2} + \frac{d}{dz}\left(\frac{1}{\rho}\right)\frac{dZ_m}{dz}\right) + \frac{1}{\rho}\left(k^2 - k_{rn}^2\right)\sum_m a_{nm} Z_m + \frac{2}{\rho}\left(k\Delta k - k_{rn}\Delta k_{rn}\right)Z_n = 0. \quad (3.5)$$

Substituting the unperturbed normal modes equation (Eq. (3.1)) into the first summation and cancelling like terms yields

$$\frac{1}{\rho}\sum_m a_{nm}\left(k_{rm}^2 - k_{rn}^2\right)Z_m + \frac{2}{\rho}\left(k\Delta k - k_{rn}\Delta k_{rn}\right)Z_n = 0. \quad (3.6)$$

Finally, multiplying by Z_n, integrating from zero to infinity over depth, and

utilizing the orthonormality of the density and eigenfunctions results in

$$a_{nn}\left(k_{rn}^2 - k_{rn}^2\right) + 2\int_0^\infty \frac{k(z)\Delta k(z) Z_n^2(z)}{\rho(z)}\, dz - 2k_{rn}\Delta k_{rn} = 0 \tag{3.7}$$

or

$$\Delta k_{rn} = \frac{1}{k_{rn}}\int_0^\infty \frac{k(z)\Delta k(z) Z_n^2(z)}{\rho(z)}\, dz. \tag{3.8}$$

Equation 3.8 requires one further simplification, the Δk term must be reduced into its components. Utilizing the relationship

$$k + \Delta k = \frac{\omega}{c_0 + \Delta c} \tag{3.9}$$

and approximating to first order results in

$$\Delta k = -k\frac{\Delta c}{c_0}. \tag{3.10}$$

Substituting into Eq. (3.8) results in

$$\Delta k_{r,n} = \frac{-1}{k_{r,n}}\int_0^\infty \frac{k^2(z)\Delta c(z)}{\rho(z)c_0(z)}\, |Z_n(z)|^2\, dz \tag{3.11}$$

which has been used by Rajan [11], Ballard [15], and several others for pertur-

bative inversion in depth only. This detail of this derivation is attributed to Poole [24]. This technique is not explored any further in this paper. Several other works have investigated and explained its implementation and the reader is directed to any of those sources [4, 11–15].

In the range-independent case, it is possible to write

$$t_n(r) = \frac{r}{c_{\mathrm{grp}_n}} \tag{3.12}$$

where $t_n(r)$ is the travel time of the mode n over distance r. Applying the relationship between wave number and group speed given in Eq. (2.20a) on Eq. (3.11) results in

$$(\Delta c_{\mathrm{grp},n})^{-1} = \frac{\partial}{\partial\omega}\frac{-1}{k_{r,n}}\int_0^\infty \frac{k^2(z)\Delta c(z)}{\rho(z)c_0(z)}\,|Z_n(z)|^2\;dz \tag{3.13}$$

where $\Delta c_{\mathrm{grp},n}$ is the difference between the "measured" and predicted group speeds of the n^{th} mode. Unfortunately, it is not possible to directly measure group speed. Equation (3.16) gives the most convenient relationship between group speed and a measurable quantity, travel time. Combining this relationship with Eq. (3.13) yields

$$\Delta t_n = \frac{\partial}{\partial\omega}\int_0^\infty \frac{-1}{k_{rn}(s,\omega)}\frac{\omega^2\Delta c(s,z)r}{\rho(s,z)c_0^3(s,z)}\,|Z_n(s,z,\omega)|^2\;dz, \tag{3.14}$$

which gives the perturbation to the modal travel time along an acoustic path.

3.2.2 The Fredholm Integral

Equation (3.14) can be written in the form of a Fredholm integral of the first kind:

$$d_n = \int_0^\infty m(z)G_n(z)\ dz \quad \text{for } n = 1, ..., N, \tag{3.15}$$

where d_j is the difference between the n^{th} measured modal travel time and the corresponding predicted modal travel time, $G_n(z)$ is the forward model, $m(z)$ is the correction to the sound speed profile at depth z, and N is the number of estimated modes. It is important to remember that this equation requires a continuous function of depth ($m(z)$). This is essentially the same as estimating the sound speed at an infinite number of depths. Since this would require an infinite number of measurements to fully define the solution, the inverse problem will always be under-determined [25].

Modifying the limits of the integral in Eq. (3.15) allows for the selection of different parts of the waveguide. This work is only concerned with the water column so the limits will always be bounded between zero and the sea floor. However, even though this work does not invert for the sediment profile, some degree of knowledge of the sediment is required for the determination of the modal eigenfunctions.

3.2.3 Extension to Three Dimensions

Extending the 1D technique into 3D requires an extension of Eq. (3.12) into a range-dependent environment. This extension can be expressed as the integral

$$t_n(r) = \int_0^r \frac{ds}{c_{\mathrm{grp},n}(s)} \tag{3.16}$$

where $t_n(r)$ is the travel time of mode n over distance r. Continuing as before, the range-dependent inversion scheme is given as

$$\Delta t_n = \frac{\partial}{\partial \omega} \int_0^r \int_0^{D(s)} \frac{-1}{k_{rn}(s,\omega)} \frac{\omega^2 \Delta c(s,z)}{\rho(s,z) c_0^3(s,z)} \left|Z_n(s,z,\omega)\right|^2 \, dz \, ds. \tag{3.17}$$

This relationship allows for 3D inversions by correlating known acoustic path locations with the sound-speed correction. Since the technique is not constrained to constant bathymetry, this means that the limits of the depth integral will vary as a function of range.

3.2.4 Discretization

To solve the inverse problem, Eq. (3.17) must be recast into the Fredholm integral of Eq. (3.15). The difference in the number of integrals between Eqs. (3.17) and (3.15) is addressed later in this section. The continuous inverse problem is approximated as

$$d_n = \sum_{p=1}^{P} \sum_{z=0}^{D_p} m_{p,z} G_{n,p,z} \tag{3.18}$$

which assumes that the sound speed profile can be approximated by a finite number of points. Applying the same procedure to Eq. (3.17) gives:

$$\Delta t_n = \frac{\partial}{\partial \omega} \sum_{p=1}^{P} \sum_{z=0}^{D_p} \frac{-r_p \Delta z}{k_{n,p}(\omega)} \frac{\omega^2 \Delta c_{p,z}}{c_{0,p,z}^3 \rho_{p,z}} \, |Z_{n,p,z}(\omega)|^2 \tag{3.19}$$

where p and P are the range and cross-range index and total element number, respectively. Additionally, r_p is the length of an acoustic path in the pth range/cross-range element. Before proceeding, the derivative operator must be applied:

$$\Delta t_n = \sum_{p=1}^{P} \sum_{z=0}^{D_p} \frac{r_p \Delta z}{k_{n,p}(\omega)} \frac{\omega^2 \Delta c_{p,z}}{c_{0,p,z}^3 \rho_{p,z}} \, |Z_{n,p,z}(\omega)|^2 \times \left(\frac{1}{k_{n,p}(\omega) c_{gr,n,p}(\omega)} - \frac{2}{\omega} - 2 \frac{\partial}{\partial \omega} \ln(|Z_{n,p,z}(\omega)|) \right) \tag{3.20}$$

Recasting this equation into Eq. 3.18 results in:

$$G_{n,p,z} = \frac{r_p \Delta z}{k_{n,p}(\omega)} \frac{\omega^2}{c_{0,p,z}^3 \rho_{p,z}} \, |Z_{n,p,z}(\omega)|^2 \times \left(\frac{1}{k_{n,p}(\omega) c_{\mathrm{grp},n,p}(\omega)} - \frac{2}{\omega} - 2 \frac{\partial}{\partial \omega} \ln(|Z_{n,p,z}(\omega)|) \right) \tag{3.21a}$$

$$d_n = \Delta t_n \tag{3.21b}$$

$$m_{p,z} = \Delta c_{p,z}. \tag{3.21c}$$

The translation from two summations to one summation is handled by stacking the information so that when the maximum depth in one block is reached, the zero depth in an adjacent block follows. Stacking is feasible because the summation over depth involves a finite number of terms. It is simpler to write Eq. (3.18) in matrix form as

$$\mathbf{d} = \mathbf{G}\mathbf{m} \tag{3.22}$$

where $\mathbf{d}$ is an $[N \times 1]$ vector, $\mathbf{m}$ is an $[M \times 1]$ vector, $\mathbf{G}$ is an $[N \times M]$ matrix, N is the number of measurements, and M is the number of discrete volume elements in the search area.

3.3 Approximate Equality Constraints

While many techniques are available to better condition an ill-posed problem, this work investigated two unique methods before settling on a single hybrid method. Without the use of these conditioning techniques, small data error can result in large deviations and a potentially infinite number of solutions. The first method explored in this paper employs approximate equality constraints, which is a combination of Tikhonov regularization (or a relative equality constraint) and an absolute equality constraint.

The use of Tikhonov regularization [26] selects a smooth solution from the previously mentioned infinite set of solutions. This is accomplished by choosing a solution that satisfies both the data and a smoothness constraint:

$$\mathbf{Lm} = \mathbf{0} \tag{3.23}$$

where $\mathbf{L}$ is a discrete version of the second order differential operator. Smoothing is applied to remove solutions with sharp discontinuities.

Other types of Tikhonov regularization can be employed by using derivatives of order other than two. For example, utilizing the first derivative will select solutions that are flat instead of smooth. The smoothing in this work is performed in the depth direction only. In other words, the smoothing operator does not approximate the Laplacian operator but $\frac{d^2}{dz^2}$ only.

The second constraint employed by this method is referred to as an absolute equality constraint and is given as

$$\mathbf{Am} = \boldsymbol{\alpha} \tag{3.24}$$

where $\mathbf{A}$ is a matrix specifying points in the model space and $\boldsymbol{\alpha}$ is a vector of preselected perturbation amount, uniformly zero for this work. This constraint allows for *a priori* knowledge of the water column to be included in the inversion result. This is done by forcing perturbations to the solution

at selected points to be restricted to a predefined amount. The constraint is primarily and necessarily employed to stabilize the solution by restricting the perturbation at the top and bottom of the water column. It is necessary to constrain the solution at the top of the water column because every mode has a node at the sea surface. This means no direct estimate of the sound speed at the sea surface is available. The constraint is applied at the bottom of the water column to account for possible aliasing from uncertainties in the sea floor [4].

3.3.1 Combination

Using Lagrange multiplication [2] to minimize Eq. (3.22) subject to Eqs. (3.23) and (3.24) yields the least-squares constrained solution. This is accomplished by defining the Lagrange function as

$$\Lambda = (\mathbf{Gm} - \mathbf{d})^2 - \lambda_1^2(\mathbf{Lm} - \mathbf{0})^2 - \lambda_2^2(\mathbf{Am} - \boldsymbol{\alpha})^2, \tag{3.25}$$

where Λ is the Lagrange function, λ_1, and λ_2 are the Lagrange multipliers for the relative and absolute equality constraints, respectively. The square is applied to each term of Eq. (3.25) so that when the Lagrange function is minimized, the resulting solution will be least-square. The Lagrange multipliers are squared for consistency, but it is entirely acceptable to redefine the equation so that the multipliers are not squared. Taking a derivative of the Lagrange function with respect to $\mathbf{m}$ and setting the result equal to zero gives:

$$\frac{\partial}{\partial \mathbf{m}}\Lambda = \frac{\partial}{\partial \mathbf{m}}\left((\mathbf{G}\mathbf{m}-\mathbf{d})^2 - \lambda_1^2(\mathbf{L}\mathbf{m})^2 - \lambda_2^2(\mathbf{A}\mathbf{m}-\boldsymbol{\alpha})^2\right) = \mathbf{0}. \tag{3.26}$$

Taking derivatives of the Lagrange function with respect to the Lagrange multipliers results in Eqs. (3.23) and (3.24).

Simplifying Eq. (3.26) and solving for $\mathbf{m}$ results in:

$$\hat{\mathbf{m}} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda_1^2\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda_2^2\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\left(\mathbf{G}^{\mathrm{T}}\mathbf{d} + \mathbf{A}^{\mathrm{T}}\boldsymbol{\alpha}\right), \tag{3.27}$$

where the hat notation is added to draw attention to the fact that the solution is an estimate of the sound-speed correction. This solution corresponds to a local minimum. This means that it is possible for the method to become "stuck" in the local minimum when an absolute minimum exists elsewhere. Replacing the value of $\boldsymbol{\alpha}$ with a vector of zeros, as discussed previously, gives the final result for this method:

$$\hat{\mathbf{m}} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{G} + \lambda_1^2\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda_2^2\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\left(\mathbf{G}^{\mathrm{T}}\mathbf{d}\right). \tag{3.28}$$

So far, no discussion has been given for the values of the Lagrange multipliers. In general, the values of the Lagrange multipliers are determined by setting derivatives of the Lagrange function with respect to other variables (besides the multipliers or the input variable $\mathbf{m}$) to zero. Doing so in this case is not

helpful because the systems equation (Fredholm integral) and the constraints have no other terms in common so the discussed procedure yields the unhelpful result of $\lambda_1 = \lambda_2 = 0$. This work has utilized the Lagrange multipliers as user-controlled inputs to set the relative strength of the constraints with respect to the systems equation.

3.4 The Covariance Method

The second method explored in this paper is accredited to Tarantola [27]. A derivation of Tarantola's technique is outside the scope of this work, the reader is instead directed to the third chapter of [27]. The covariance method does not utilize approximate equality constraints. Instead, the method employs *a priori* estimates of the data and model covariance. Specifically,

$$\hat{\mathbf{m}} = \left(\mathbf{G}^{\mathrm{T}}\mathbf{C}_D^{-1}\mathbf{G} + \mathbf{C}_M^{-1}\right)^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{C}_D^{-1}\mathbf{d} \tag{3.29}$$

where $\mathbf{C}_D$ is the data covariance matrix, $\mathbf{C}_M$ is the model covariance matrix, and the other variables remain as previously defined. In addition to providing an estimate for the sound-speed correction this method provides an estimate to the inverted sound-speed covariance:

$$\hat{\mathbf{C}}_M = \left(\mathbf{G}^{T}\mathbf{C}_D^{-1}\mathbf{G} + \mathbf{C}_M^{-1}\right)^{-1}. \tag{3.30}$$

This method is similar to the method of approximate equality con-

straints in that there are two variables that must be accounted for prior to beginning the inversion. However, unlike the method of approximate equality constraints, the covariance method utilizes physically recognizable inputs. Additionally, the covariance method differs in that the variables are matrices whereas in the method of approximate equality, the predetermined variables are scalars.

The simplest estimations of $\mathbf{C}_D$ and $\mathbf{C}_M$ are as identity matrices times a constant. In the case of $\mathbf{C}_D$ this estimate is best made as some percentage of the average travel time of all paths. For $\mathbf{C}_M$ this estimate should be the expected sound-speed variability. Application of these two covariance matrices proves to be inadequate. This is not entirely unexpected, as it would mean the sound speed in each discrete element is independent of the surrounding elements. Similarly, it requires that the error along each acoustic path is constant regardless of the path length.

A more complex estimate of the data covariance matrix can be made by acknowledging that each value along the diagonal corresponds to a specific acoustic path. This realization allows for each *a priori* covariance estimate to be proportional to its individual acoustic path. This approximation assumes that each frequency and mode along their geometric path is independent. A better estimate of the *a priori* model covariance requires some knowledge of the covariance shape instead of just an estimate of the magnitude of the variance. There are several possible covariance distributions worth exploring. Rajan, for example, utilizes a Gaussian distribution to determine the covariance shape

[11]. This work will utilize the shape generated from the method of approximate equality in conjunction with an identity matrix. Namely,

$$\mathbf{C}_M = C_{M0}\left(\left(\lambda_1^2\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda_2^2\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1} + \mathbf{I}\right) \tag{3.31}$$

where $\mathbf{L}$ and $\mathbf{A}$ are as previously defined, $\mathbf{I}$ is an identity matrix of size [MxM] and C_{M0} is a normalized coefficient. Specifically, C_{M0} is given as:

$$C_{M0} = \frac{C_V}{\max\left(\left(\lambda_1^2\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda_2^2\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1} + \mathbf{I}\right)} \tag{3.32}$$

where C_V is the expected sound-speed variability. This scales the estimate of $\mathbf{C}_M$ so that its maximum value is the expected variability and that the other values vary based upon the combined effects of the approximate equality constraints and the identity matrix. The Lagrange multipliers are used to weight the approximate equality constraints relative to the identity matrix. The final result is best thought of as a combination of two distinct distributions: a Dirac delta function and a smoothing function and is given as

$$\hat{\mathbf{m}} = \hat{\mathbf{C}}_M\mathbf{G}^{\mathrm{T}}\mathbf{C}_D^{-1}\mathbf{d} \tag{3.33}$$

where

$$\hat{\mathbf{C}}_M = \left(\mathbf{G}^{\mathrm{T}}\mathbf{C}_D^{-1}\mathbf{G} + C_{M0}^{-1}\left(\left(\lambda_1^2\mathbf{L}^{\mathrm{T}}\mathbf{L} + \lambda_2^2\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1} + \mathbf{I}\right)^{-1}\right)^{-1}. \tag{3.34}$$

These two equations will be utilized throughout the remainder of this work. While $\mathbf{C}_D$ and C_{M0} have physical significance and can be approximated for a given data set and search volume, λ_1 and λ_2 are more difficult to determine. The best method for determining the two Lagrange multipliers is empirical. Fortunately, once the values have been set for a known test case, the same values can be utilized for other inversions.

3.5 Resolution

As mentioned previously, averaging results from solving the Fredholm integral (Eq. (3.15)). This is a result of attempting to resolve a continuous function with a finite amount of data. More precisely, the continuous inverse problem is the limit of a discrete inverse problem where the number of parameters is infinite and will always be under-determined [25]. As a result, it is impossible to determine $m(r)$ exactly at $r = r_0$, instead, the inverted parameters are local averages of the true values.

According to Backus and Gilbert [28], the averaged parameters can be related to the true parameters through a simple relationship given as

$$\hat{m}_{\vec{r}_0} = \int_V R(\vec{r}_0, \vec{r}) m(\vec{r})\, dV \tag{3.35}$$

where $R(\vec{r_0}, \vec{r})$ is an averaging or resolving kernal and $\vec{r}/\vec{r_0}$ are vector coordinates. Ideally, $R(\vec{r_0}, \vec{r})$ would be a delta function, indicating that the estimated

parameter is in fact, the true value. Therefore, an estimate of the resolving power of the inversion can be made by determining how well the averaging kernel approximates a delta function in some least squares sense. While additional measures of Dirac delta functions are available, none will be explored here. Additional information can be found in Backus and Gilbert's pioneering paper [28].

To determine the resolution kernel, the method of Lagrange multiplication will again be employed. Before beginning, several additional equations are required. Since the inversion is a linear process, the averaged model parameter at any location can be written as a linear combination of the data, namely,

$$\hat{m}_{\vec{r}_0} = \sum_{i=1}^{N} a_i(\vec{r}_0) d_i(m). \tag{3.36}$$

Utilizing a combination of Eqs. (3.15), (3.35), and (3.36) results in

$$R(\vec{r}_0, \vec{r}) = \sum_{i=1}^{N} a_i(\vec{r}_0) G_i(\vec{r}). \tag{3.37}$$

Equation (3.37) is sufficient to determine an estimate of the averaging kernal. The rest of this derivation will be performed in 1D, namely, $\vec{r} = (0, 0, z)$ will be given as z. The final result can be easily extended into 3D. Since the delta function only has real meaning inside an integral, it is necessary to define the Lagrangian as

$$\Delta = \int_0^D \left(\left(\sum_i a_i \left(z_0 \right) G_i \left(z \right) \right) - \delta \left(z - z_0 \right) \right)^2 dz. \tag{3.38}$$

To begin, it is necessary to expand the square

$$\Delta = \int_0^D \left(\left(\sum_i a_i \left(z_0 \right) G_i \left(z \right) \right)^2 - 2 \left(\sum_i a_i \left(z_0 \right) G_i \left(z \right) \right) \delta \left(z - z_0 \right) + \delta^2 \left(z - z_0 \right) \right) dz \tag{3.39}$$

and simplify the integral

$$\Delta = \int_0^D \left(\sum_i a_i \left(z_0 \right) G_i \left(z \right) \right)^2 dz - 2 \left(\sum_i a_i \left(z_0 \right) G_i \left(z_0 \right) \right). \tag{3.40}$$

Finally, it is necessary to take the derivative of the Lagrangian with respect to the linear coefficients:

$$\frac{d\Delta}{da} = 2 \sum_i \int_0^D a_i \left(z_0 \right) G_i \left(z \right) dz \sum_j \int_0^D G_j \left(z \right) dz - 2 \sum_i G_i \left(z_0 \right) = 0. \tag{3.41}$$

Recognizing that the continuous functions must be discretized and can be conveniently cast as a matrix equation results in an expression for a

$$\mathbf{a} \left(z_0 \right) = \left(\mathbf{G} \mathbf{G}^{\mathrm{T}} \right)^{-1} \mathbf{G} \left(z_0 \right) \tag{3.42}$$

where

$$\int_0^D G_i(z)\, G_j(z)\, dz = \mathbf{G}\mathbf{G}^{\mathrm{T}}. \tag{3.43}$$

Inserting this expression into Eq. (3.37) results in a relationship between the resolution matrix and the forward model,

$$\mathbf{R} = \mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\mathbf{G}^{\mathrm{T}}\right)^{-1}\mathbf{G}. \tag{3.44}$$

The resolution matrix is generally not invertible. If it were invertable, it would be possible to recover the exact solution from the estimate.

The resolution matrix is particularly useful when applied in range and cross-range only. This application highlights which range-independent blocks will have unique, non-unique, or nonexistent inversion results. In other words, the 2D resolution matrix illustrates where the inversion will have good results based solely on geometric paths. Another helpful metric for visualization is the resolution length and is given by

$$\mathbf{R}_i = \frac{\sum_{j=1}^{N} \mathbf{R}_{ij}^2}{\mathbf{R}_{ii}^2}\Delta x \Delta y \tag{3.45}$$

The term resolution "length" is historically from 1D problems but will be used

in this document regardless of the number of inversion dimensions. Equation (3.45) establishes a 2D resolution length. Determination of the 3D resolution length would require the 3D resolution matrix and the inclusion of the depth increment.

Both the resolution matrix and the resolution length are calculated from the forward model and only describe the resolving power of the data without the inclusion of the previously discussed *a priori* information or data noise. The spread of the *a priori* model covariance will result in an increase in the amount of averaging from what is accounted for in the resolution matrix. The resolution matrix and lengths are still useful indicators of how well resolved the final inversion will be before the inversion is performed.

3.6 Summary

In this chapter, a linearized perturbative inverse scheme relating acoustic travel time measurements to sound speed values in a 3D grid was developed. Two methods of stabilizing and selecting a solution were introduced. The first method was referred to as the method of approximate equality and utilized a combination of second order Tikhonov regularization and the absolute equality constraint. The second method was referred to as the covariance method and employed *a priori* statistical knowledge of the environment and data. A final solution amalgamated the two stabilization methods. This allowed for the inclusion of physically recognizable *a priori* variance information with an appropriate covariance spread.

Finally, an expression for the solution variance and resolution was presented. The method for estimating variance is an improvement over previous work as Monte Carlo methods [26] are no longer necessary. Instead, the variance is determined in step with the sound-speed correction. A method for determining resolution was presented according to the definition of Backus and Gilbert [28]. The resolution matrix allows for predictions of the resolving power of the data before the inversion has been performed. Amongst other things, this information could be used to determine acoustic source and receiver placement.

Chapter 4

Three-Dimensional Sound-Speed Inversion Algorithm: Development and Application

4.1 Introduction

This chapter explores the development and implementation of the inversion technique described in Chapter 3. The Matlab code necessary to perform an example inversion can be found in the Appendix. So that an easy comparison can be made between the inversion results and the true sound speeds, this chapter only utilizes simulated data. Additionally, since the focus of this work is solely on the sound speed in the water, the correct sediment sound speed will be assumed when performing the inversion.

The chapter is primarily split into two parts. The first part utilizes entirely simulated environmental parameters. It begins with a 1D example and explores how robust the technique is when faced with different types of errors. Next, a 3D example is explored with additional focus on the *a priori* covariances and resolution matrices.

The second part utilizes more a sophisticated environmental simulation to approximate environmental parameters on the Hudson Shelf [29]. This allows for high range and cross-range resolution of the water-column sound-

speed field when generating modal travel times. This data set provides a good approximation of the variability that would be expected in the region.

4.2 One-Dimensional Inversion

It is convenient to begin with a 1D example as it is simpler to visualize the inversion output without the complexity of variations in other dimensions. Additionally, by keeping the environment simple, it is more likely that an intuitive understanding can be developed. To that end, a single acoustic path will be utilized for the majority of this section.

The inversion algorithm has been constructed under the assumption that the search area can be approximated as a Cartesian grid (where the search area is the range and cross-range region of the ocean in which the inversion will be performed). As such, latitude and longitude coordinates are translated onto a Cartesian grid. Unless otherwise stated, all travel time data will be noise free and will ignore the effects of horizontal refraction.

In the case of the 1D inversion, determining the modal travel times is fairly simple. The entirety of this work utilizes KRAKEN to determine the group speed given the environmental properties. KRAKEN calculates normal modes in the shallow ocean regardless of the complexity of the water-column and sediment sound speeds or densities. An in-depth explanation of the intricacies of the program can be found online [30].

When the group speed is known, the travel time can be determined

Table 4.1: Single source and receiver coordinates for 1D inversion

	Cross-Range Coordinate (km)	Range Coordinate (km)
Source	71.95	32.83
Receiver	10.40	8.96

using Eq. (3.16). Table 4.1 gives the range and cross-range coordinates for the source and receiver. Source and receiver locations are critical to utilize this technique.

Figure 4.1 shows the water-column sound-speed profile which will be used for this example. Additionally, the example utilizes an assumed water-column density of 1 kg/m^3, and a three layer sediment where each layer is sixteen meters thick and has sound speeds of 1550, 1600, and 1650 m/s from the seafloor down, each with a density of 1.6 kg/m^3. The water-column is 74 m in depth. All examples utilize three acoustic frequencies: 50, 100, and 150 Hz. The number of modes varies based on the depth and water-column sound speed, in this case, three, five, and eight modes are used. The group

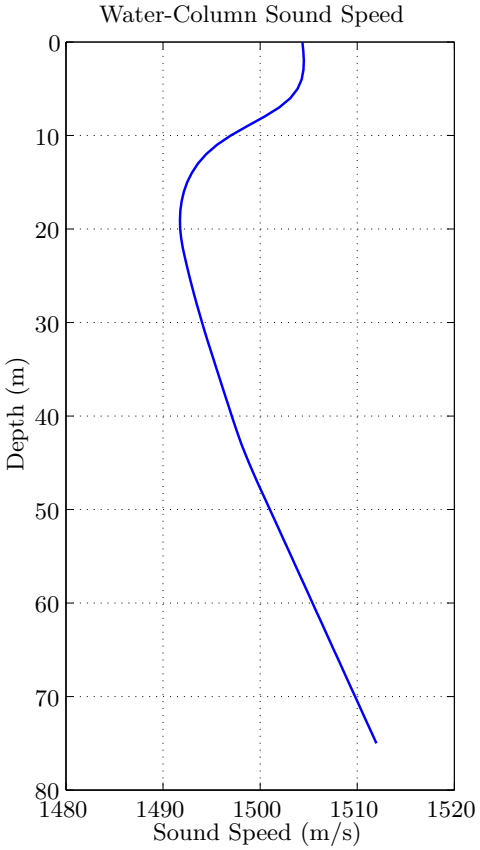


Figure 4.1: 1D water-column sound speed

speeds for each mode of each frequency are given in Table 4.2.

Table 4.2: Actual group speeds (m/s) of all propagating modes in 1D example

Mode Number	50 Hz	100 Hz	150 Hz
1	1487.8	1491.2	1492.0
2	1453.6	1485.1	1490.7
3	1560.7	1463.1	1482.2
4		1465.5	1466.1
5		1474.7	1461.4
6			1489.0
7			1431.7
8			1524.4

The background sound-speed profile from which the inversion begins is taken as the mean sound speed. In practice this information would be unavailable necessitating the use of some other source of information such as historical data or using a Monte Carlo search for the best beginning profile.

It is generally a good idea to perform a Monte Carlo search as the absolute constraint can cause the background profile to be overly dominant if a poor starting profile was used. To perform the search, a range of expected sound speeds must be determined. The inversion is then performed from a number of starting profiles within the range. The starting profile is chosen according to the solution which has the smallest residual.

Before performing the inversion it is necessary to construct the *a priori* model covariance as per Section 3.4. Figure 4.2 illustrates the resulting covariance matrix (with units of m^2/s^2) where Lagrange multipliers are $\lambda_1 = 500$

and $\lambda_2 = 0.1$ respectively. These values were arrived at by choosing the desired relative strength of each condition and iterating until that result was achieved. The value of C_V is approximately 44 m^2/s^2. $\mathbf{C}_D$ is constructed by assuming the standard deviation of each travel time estimate is equal to 1% of the estimate. These values are used for every analysis performed in this work.

Examination of Figure 4.2 shows decreasing correlation as the distance between points increases so that the top and bottom are uncorrelated. Additionally, there is a slight emphasis on the diagonal. This result is exactly as expected, desired, and designed.

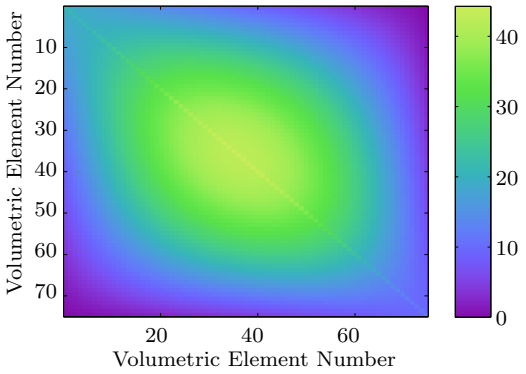


Figure 4.2: 1Dl *a priori* model covariance matrix

4.2.1 One-Dimensional Results

This section discusses the results of the 1D inversion under the environmental and experimental setup discussed in the previous section. Afterwards, the same error-free example is presented with additional spatial sampling. Finally, the single acoustic path example is discussed with different types of errors.

Figure 4.3 shows the results of the inversion. In the figure, the red line corresponds to the background profile, the blue to the actual sound-speed profile and the black line to the inversion result. The magenta and green lines

correspond to the 95% confidence interval around the solution. The magenta line corresponds to the *a priori* estimate (related to the diagonal in Figure 4.2). The green lines correspond to the estimated inversion covariance from Eq. (3.30). This legend will be used for all range-independent results.

There are two desirable features to look for in each inversion result: first, that the inversion result tends to the actual profile, second, that the confidence intervals have narrowed. In this example, both of these trends have held. The inversion has diverged from the background profile and is more representative of the truth than before the inversion began. The *posteriori* confidence interval is considerably narrower than the *a priori* confidence interval. The RMS difference between the inversion result and the actual profile is 1.86 m/s. This value provides a baseline from which all 1D results can be compared.

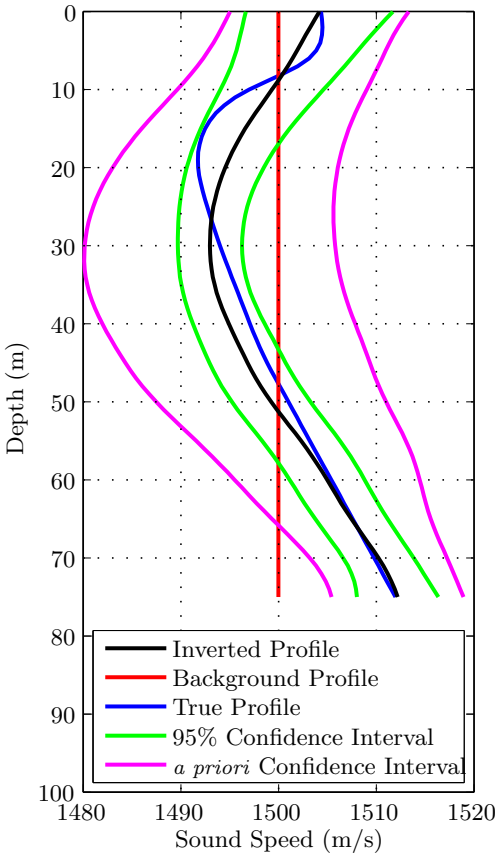


Figure 4.3: 1D range-independent inversion results with single source and single receiver

Additionally, a more in depth examination shows the effects of the constraints placed on the *a priori* model covariance. Namely, the inversion does

not capture the sharp changes in the sound-speed profile. This is apparent at approximately five and fifteen meters in depth where the actual profile experiences a large change over a small depth interval but the inversion compensates by initiating the change over a larger interval. This is a result of the smoothing constraint—the constraint is too heavily weighted relative to the inversion kernel (G) to allow for the sudden change. Similarly this result shows that the inversion tends to overestimate and underestimate the solution at different depths. As a result, the average difference between the inversion and the actual profiles is low. This trend of over and under estimation averaging out will show up repeatedly in all three dimensions.

A more detailed examination of the confidence interval also shows some of the desired trends. The *a priori* confidence interval width is greatest in the center, but the *posteriori* confidence interval width is greatest at the ocean surface and bottom. The *posteriori* trend makes sense, since the majority of the information in the inversion is in the center of the wave guide. As mentioned previously in Chapter 2, all modes have a node at the sea surface and begin decaying in the sea floor. This means that the greatest uncertainty about the result is at the top and bottom of the water-column as the information at those locations is largely derived from the smoothing and absolute constraints instead of the acoustic data.

To explore the effects of spatial sampling, an additional source and receiver were added to the experimental setup and the inversion was rerun. Table 4.3 gives the coordinates of the sources and receivers. Since the en-

Table 4.3: Multiple source and receiver coordinates for 1D inversion

	Cross Range Coordinate (km)	Range Coordinate (km)
Sources	71.95	32.83
	50.91	45.46
Receivers	10.40	8.96
	55.74	23.53

vironment is unchanged from the previous case, the group speeds shown in Table 4.2 remain unchanged.

Figure 4.4 shows the results from the 1D inversion with four times the number of acoustic paths as before. The results show that the inversion is more representative of the actual profile in this case. Additionally, the *posteriori* confidence intervals are narrower. The RMS difference between the inversion with four transducers and the truth is 0.91 m/s. This represents over a 50% improvement from the single source/single receiver case.

A comparison between the results of the four-path inversion and the one-path inversion shows that the four-path result is capable of capturing the changes over a smaller interval than the one-path inversion. This means the additional information provided by increased spatial sampling has reduced the importance of the *a priori* covariance estimate so that the smoothing constraint is less dominant. However, the smoothing constraint remains dominant at the top of the water column. This will always be the case as there will never be any information in the modal travel times about the top of the water column. Similarly, the confidence interval can never decrease beyond a certain point at the sea surface for the same reason.

Since the data for the one and four-path examples are error free, the additional paths in the second example are redundant. The improvement seen in Figure 4.4 from Figure 4.3 is caused by an increase in the weight of the inversion kernel over the *a priori* model covariance. However, if random noise were present in the examples, the additional paths would help to reduce the error caused by the noise.

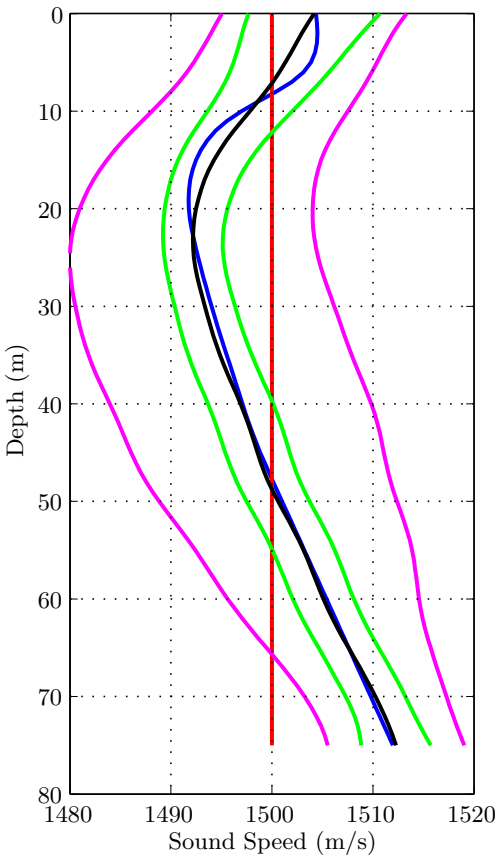


Figure 4.4: 1D range-independent inversion results with two sources and two receivers (See Figure 4.3 for legend)

4.2.2 Sensitivity to Error

This section investigates the effects of different types of errors in the inversion. It is worthwhile to investigate the effects of these errors to aid in diagnostics in applied inversions. The error case studies are performed in 1D because visualization is most convenient. A single acoustic path is utilized for the same reason. Errors will be considered on three separate parameters. The first parameter with added error is the source/receiver locations. The second parameter with added error is the modal travel times. The third parameter with added error is the ocean depth.

It is possible to investigate the effects of errors in other parameters such as frequency, mode number, or sediment sound speeds. Errors in frequency and mode number are not explored. An investigation of the effects of sediment uncertainty has been performed by others [4] and will not be repeated here.

4.2.2.1 Source/Receiver Location Errors

This section explores the effect of errors in estimated source/receiver position. Since this analysis is conducted in a range-independent environment with a single acoustic source and receiver it is more convenient and equivalent to perform the error analysis on acoustic path length instead of range and cross-range coordinates. However, in application, this path length data comes from source/receiver location and it is that information that has an attached uncertainty.

The level of uncertainty depends on the experimental setup. In the case of the acoustic swarm (free-floating acoustic transducers), uncertainty depends on global positioning satellite (GPS) signal strength and available satellite number; in the case of towed arrays will depend on the length of the tow cable, cruise speed, array weight, GPS considerations, etc.; and in the case of moored arrays it depends on placement accuracy and GPS considerations. This analysis is not sensitive to the experimental setup and utilized an over- or under-estimated acoustic path length in the inversion. The correct path length was utilized in generating the modal travel time data.

Figure 4.5 shows the results from the four different inversions with er-

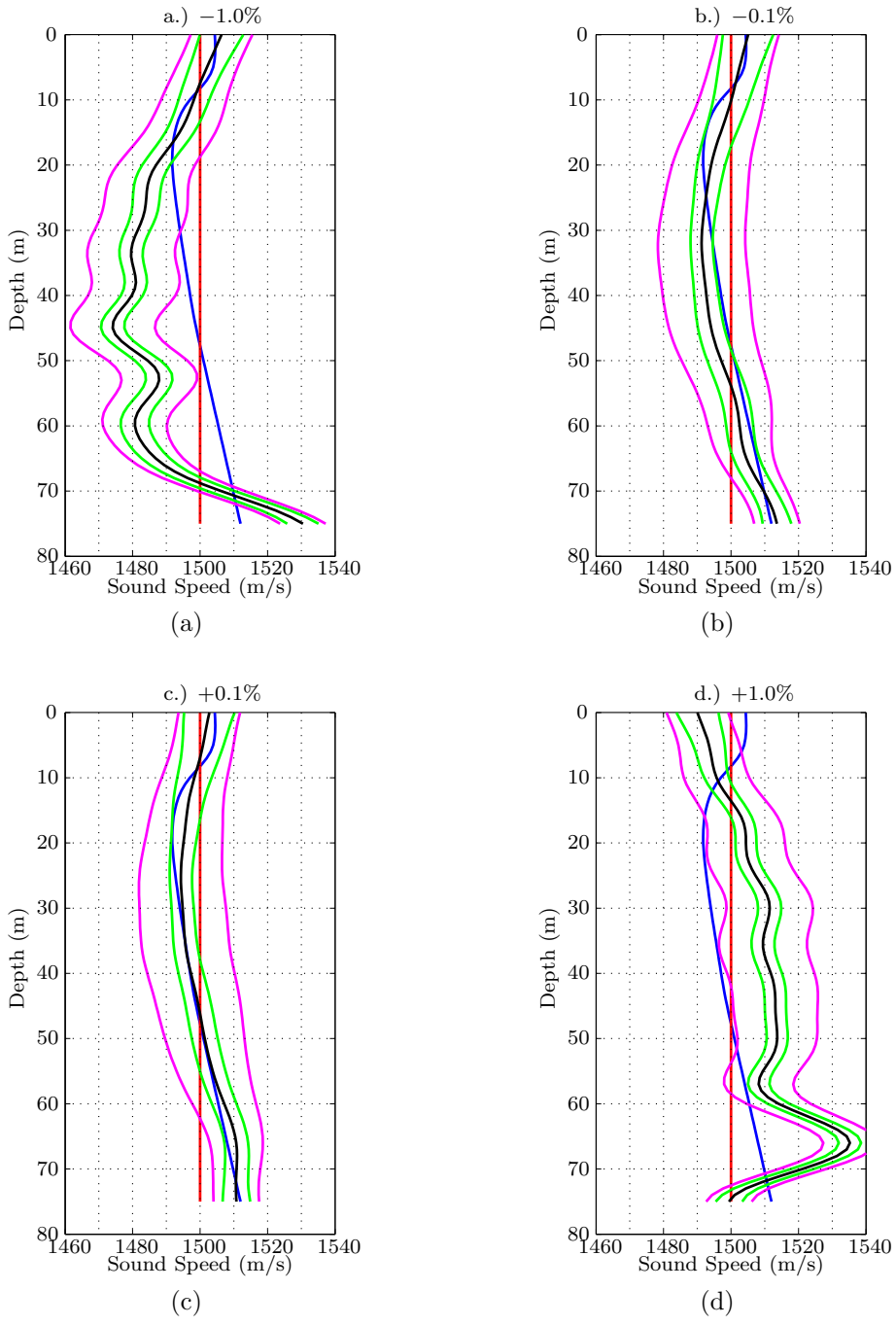


Figure 4.5: Range-independent inversion results with errors in path length, assumed path shorter than truth (-1.0% and -0.1%) and assumed path longer than truth (+0.1% and +1.0%) (See Figure 4.3 for legend)

rors in path length. The RMS error in Figure 4.5(a) (1.0% short) is 14.53 m/s, which corresponds to a 681% increase in error. The RMS error in Figure 4.5(b) (0.1% short) is 3.03 m/s, which corresponds to a 63% increase in error. The RMS error in Figure 4.5(c) (0.1% long) is 1.89 m/s, which corresponds to a 2% increase in error. The RMS error in Figure 4.5(d) (1.0% long) is 13.60 m/s, which corresponds to a 632% increase in error.

Since the path length is approximately 66 km, none of these errors are realistic under normal operating conditions. More realistic errors would be on the order of 10 m at most which corresponds to an error of approximately 0.01%. However, this level of mismatch is helpful to illustrate the effect of this type of error. As expected, if the path is short, the estimated sound speed is slower and if the path is long, the estimated sound speed is faster.

Figure 4.5(b) and Figure 4.5(c) show that errors on the order of 0.1% maintain the same shape as the actual sound speeds and that the truth is encompassed by the error bars. Additionally, the difference between the inversion and the actual profile is still on the same order as the error-free case. Since the maximum uncertainty in typical operating conditions is on the order of 10 m, this requires that any experiment utilize acoustic paths of at least 10 km in length so that the error will remain under 0.1%. Fortunately, this length is shorter than the minimum length from which modal travel times can realistically be separated in measured data. As a result, this type of error is largely unimportant.

The previous statements were all made under the assumption of straight

acoustic paths. In the event that horizontal refraction is non-negligible, the assumption of straight paths could lead to results similar to those in Figure 4.5(a). However, the problem of horizontal refraction is considerably more complex and is more appropriately discussed in a 3D setting. Mention is made here since no matter the level of range and cross-range variability, a single source and receiver pair are only capable of producing a 1D inversion.

In 3D, uncertainties in source and receiver location can lead to additional errors as it could appear that different range and cross-range blocks (range-independent regions) were being sampled than were truly being examined. In practice this primarily manifests when the acoustic path crosses through a block very near one of its corners. In this case, an error in source/receiver location can cause the path to look like it is cutting through a different blocks corner than it really is. However, this would require that the path lengths in those blocks be short relative to the total path length and as a result, would likely be short compared to the total length from all paths through the block. Therefore, it is unlikely that cutting corner errors would cause serious problems.

4.2.2.2 Data Errors

The next type of error explored utilized the addition of normally distributed random noise added to the modal travel time data. This type of error does not correspond to any single physical error, but can be used as an approximation of a combination of errors such as ambiguity in reading modal

arrivals in time-frequency filtering, internal clock mismatch between sources and receivers leading to discrepancies in launch and arrival times, or mild cases of horizontal refraction.

This type of error does not correspond to the traditional signal to noise ratio (SNR) issues typically associated with acoustic signals. Sufficiently low SNR can lead to travel time errors by increasing the ambiguity of the arrival time, but this error section does not address issues with noisy time series measurements.

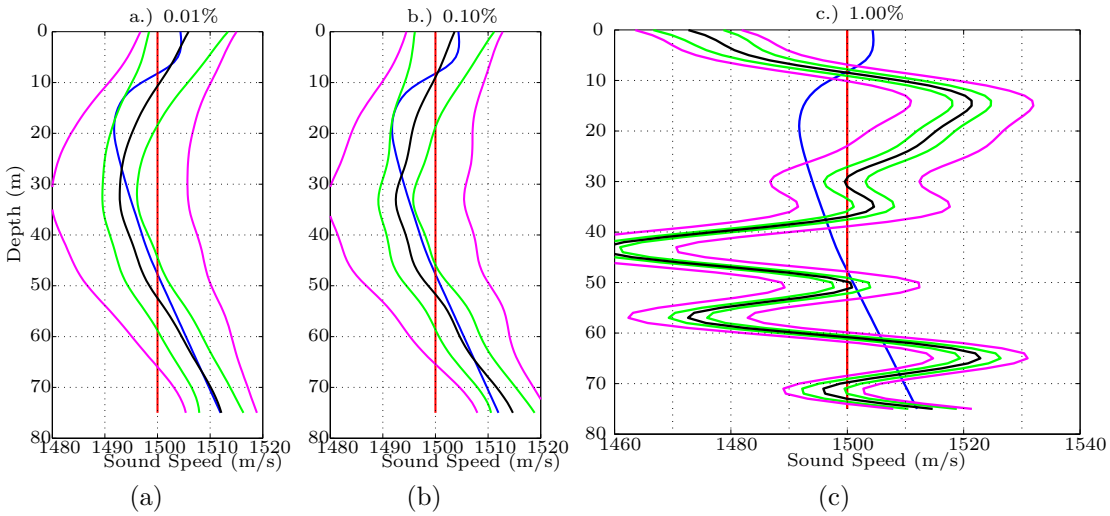


Figure 4.6: Range-independent inversion results with normally distributed random travel time errors of zero mean and standard deviations (a) 0.01%, (b) 0.10%, and (c) 1.00% of actual travel time (See Figure 4.3 for legend)

Figure 4.6 shows the results from the three different inversions with normally distributed random modal travel times errors. The RMS error of Figure 4.6(a) (0.01%) is 2.26 m/s, which corresponds to a 22% increase in solution error. The RMS error of Figure 4.6(b) (0.10%) is 2.45 m/s, which

corresponds to a 32% increase in solution error. The RMS error of Figure 4.6(c) (1.00%) is 19.39 m/s, which corresponds to a 943% increase in solution error.

The results in Figure 4.6(c) show considerable deviations from the error free results at any given depth, but the overall trend appears to be centered around the true profile. This result makes sense as the mean of the normally distributed noise was zero.

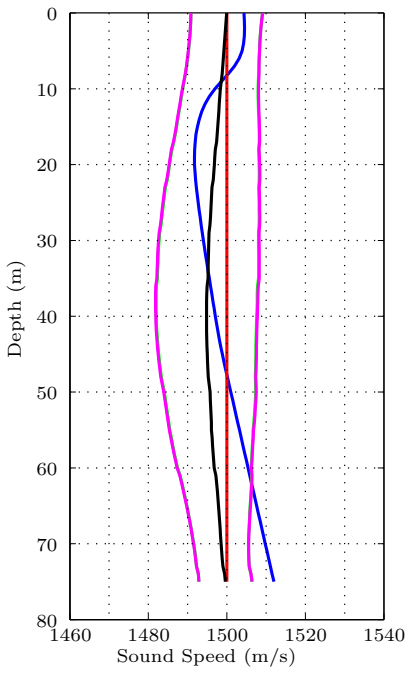


Figure 4.7: Range-independent inversion results with normally distributed random travel time errors of zero mean and standard deviation 1.00% of actual travel time with modified *a priori* data covariance (See Figure 4.3 for legend)

These results indicate that an inversion solution with high variations should have the *a priori* data covariance matrix recalculated so that the standard deviation is higher. This will allow the smoothing and absolute constraints to exert additional influence on the solution generating in a more stable result. Such a modification will result in an increase in the width of the confidence intervals. The RMS error in this case is 6.19 m/s, which corresponds to a 233% increase in error over the noise-free case. This increase is considerably less than the results prior to the modification to the data covariance matrix.

Figure 4.7 shows the results from an inversion with the same level of random noise as Figure 4.6(c) (zero mean and 1% standard deviation) with

a modified *a priori* data covariance matrix. In this case, the data covariance matrix has been reduced to an assumed 10% standard deviation of actual travel time.

Figure 4.7 shows that the inversion result is considerably more stable than in Figure 4.6(c), but that the inversion confidence intervals have not diverged from the *a priori* estimate. This lack of divergence is why the inversion confidence interval does not appear to show up in the figure—it is underneath the *a priori* interval. As mentioned previously, this is because the inversion in this figure is dominated by the constraints. This dominance is also why the inversion result has not diverged from the background profile at the top and bottom of the water column: the absolute constraint is dominating the result at those locations. Additionally, the number of iterations needed to achieve the given inversion result is dramatically increased from what was required for the previous results (6 iterations to 450 iterations). Considerably more iterations are needed because the inversion kernel (G) has little impact on the result of any given inversion so more perturbations are required to see notable changes.

4.2.2.3 Depth Inaccuracies

In practice, bathymetry is often looked up in oceanographic databases. However, a comparison of the bathymetry in a given area between two different databases can show considerable disagreement. This section explores the error in the inversion that results from faulty assumptions about waveguide depth.

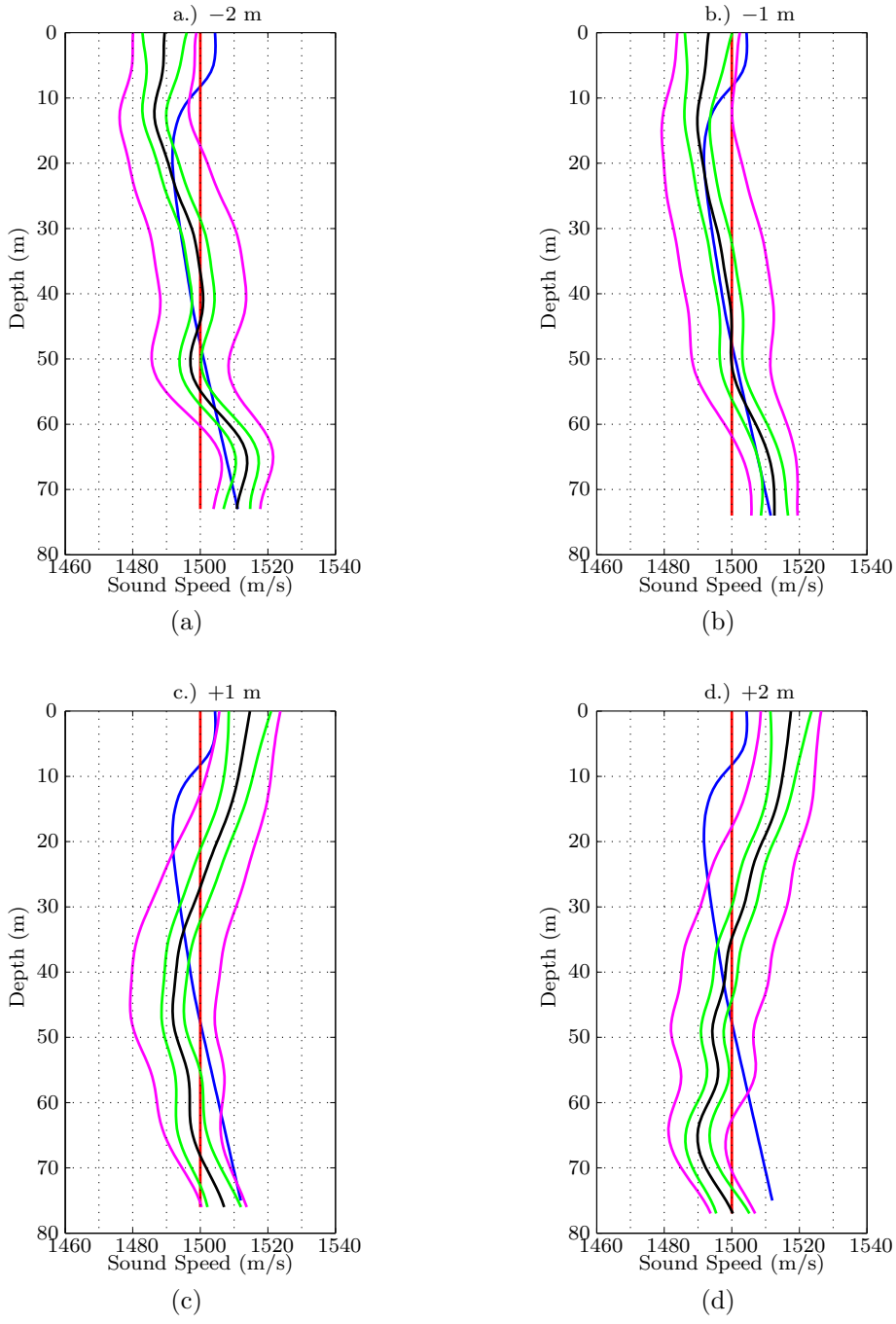


Figure 4.8: Range-independent inversion results with errors in bathymetry of (a) −2 m, (b) −1 m, (c) +1 m, and (d) +2 m (See Figure 4.3 for legend)

Figure 4.8 shows the results from the four different inversions with errors in estimated water-column depth. The RMS error in Figure 4.8(a) (2 m shallow) is 6.42 m/s, which corresponds to a 245% increase in error. The RMS error in Figure 4.8(b) (1 m shallow) is 4.47 m/s, which corresponds to a 140% increase in error. The RMS error in Figure 4.8(c) (1 m deep) is 9.29 m/s, which corresponds to a 400% increase in error. The RMS error in Figure 4.8(d) (2 m deep) is 13.01 m/s, which corresponds to a 600% increase in error.

The error in the results of Figure 4.8 is a result of biasing from assumed waveguide depth. Shallower waveguides have group speeds with slower than deeper waveguides. This can be seen mathematically for the ideal waveguide from Eq. (2.20b), repeated here for convenience:

$$c_{\mathrm{grp},n} = c_0 \sqrt{1 - \left(\frac{2n-1}{4} \frac{c_0}{fD} \right)^2}.$$

Decreasing expected depth results in a decrease in expected group speed. Similarly, increasing expected depth results in an increase in expected group speed. Since modal travel time estimates remain unchanged in this analysis, the result of errors in depth correlates exactly with the inversion bias.

4.3 Three-Dimensional Inversion

This section explores a 3D inversion. This inversion is performed on a completely simulated volume of the shallow ocean. It is helpful to think of this volume of ocean as a 2D grid of y-coordinates called the range and x-coordinates called the cross-range. Then, each discreet element in this 2D grid has a depth component (z-coordinate) that is essentially a range-independent block. Recasting the shallow ocean volume into an array of range-independent blocks helps to break down the results and processes into a familiar and readily understood region. It also makes it simpler to imagine a variable bathymetry—maintaining the mentality of a large volume implies the region is a rectangular prism which is overly simple.

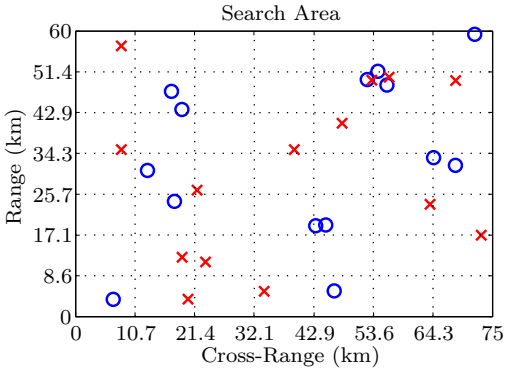


Figure 4.9: Top down view of 3D search area with sources (blue circles) and receivers (red crosses)

For this example, the search area (range and cross-range) is discretized into a 7x7 grid of 49 unique depth-dependent sound-speed profiles. The experimental setup utilizes 14 sources and 14 receivers deployed as an acoustic swarm. Since the number of acoustic paths (169) is greater than the number of blocks (49) the majority of the discretized search area is over-defined in range and cross-range. Figure 4.9 shows the locations of the acoustic sources (blue circles) and acoustic receivers (red crosses) in the search area. The coordinates

of these acoustic transducers are listed in Table 4.4.

Table 4.4: Source and receiver coordinates for simulated 3D inversion

	Cross Range Coordinate (km)	Range Coordinate (km)
Sources	56.00	48.64
	17.77	24.23
	71.80	59.31
	46.52	5.40
	45.02	19.26
	12.95	30.68
	6.78	3.64
	19.14	43.54
	64.39	33.39
	68.33	31.76
	52.47	49.80
	54.39	51.53
	17.24	47.34
	43.20	19.07
Receivers	33.92	5.32
	56.42	50.30
	8.24	35.08
	8.23	56.87
	20.24	3.66
	39.35	35.08
	72.95	17.11
	53.28	49.66
	23.39	11.46
	21.86	26.55
	63.78	23.60
	68.37	49.59
	47.95	40.61
	19.15	12.46

With the exception of the water-column sound speed and the depth (which varies between 70 and 75 m) all other environmental parameters are

unchanged from the 1D examples.

4.4 Data Generation

The data for the 3D case is generated in a similar manner to that in the 1D case with the extra caveat that the process must be repeated for each block in the path. The total travel time along the path is then the sum of the travel times through each block. This is given mathematically as

$$t_{f,m,p} = \sum_{bx} \sum_{by} \frac{\Delta p_{b_x,b_y}}{c_{\mathrm{grp}_{f,m,b_x,b_y}}} \tag{4.1}$$

where $t_{f,m,p}$ is the travel time along path p for the m^{th} mode of the f^{th} frequency, b_y is the range block number, b_x is the cross-range block number, Δp is the distance of the path through a particular block, and c_{grp} is the group speed in the block. The notation (b_x,b_y) will be used to reference specific blocks.

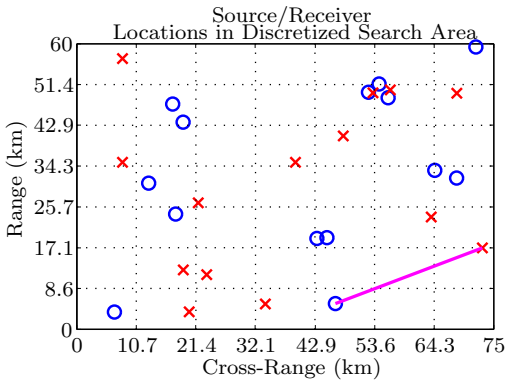


Figure 4.10: 3D source/receiver locations with emphasized acoustic path

The travel time for a particular frequency and mode along the magenta path in Figure 4.10 is given exactly as

$$t = \frac{\Delta p_{5,1}}{c_{\mathrm{grp},5,1}} + \frac{\Delta p_{6,1}}{c_{\mathrm{grp},6,1}} + \frac{\Delta p_{6,2}}{c_{\mathrm{grp},6,2}} + \frac{\Delta p_{7,2}}{c_{\mathrm{grp},7,2}} \tag{4.2}$$

where the subscripts indicate the cross-range block number and the range block number respectively. The distance in block $(6, 1)$ is very short, as a result, the information available in the block is inadequate to provide an estimate of sound speed. This calculation must be repeated for each mode of each frequency along each path.

In application, the cluster of acoustic transducers around the top-left corner of block $(5, 6)$ would be too tightly grouped to differentiate between modal arrival times. As discussed earlier, this technique essentially uses measures of dispersion to quantify a key property of the waveguide. Unfortunately, the lengths between sources and receivers in this cluster is at the limit of where modal travel times can reasonable be distinguished. However, since this example is purely simulated, the limitation on minimum path length will be ignored.

As in the 1D examples, before the inversion can be performed, it is necessary to construct the *a priori* model covariance matrix. Figure 4.11 shows the *a priori* model covariance for the 3D case. This model covariance is essentially the 1D model covariance matrix stacked along the diagonal. This result emphasizes the fact that smoothing is performed in depth only. If smoothing were performed in range or cross-range there would be additional off-diagonal terms.

This work will utilize the same background profile for each block in the region. These background profiles are determined by assuming a depth-dependent measurement is available in one of the blocks (in practice such a measurement can be obtained through CTD chains or XBTs) and taking the average sound speed. It is possible to use fits to the "measurement" of order higher than zero, but it is generally recommended to restrict the fit to order one or zero in case the fit is in a shallow block so that errors from extrapolation in deeper blocks are negligible. In practice, other methods for determining background profile can be used such as historical information about the region, but this will not be investigated in this work.

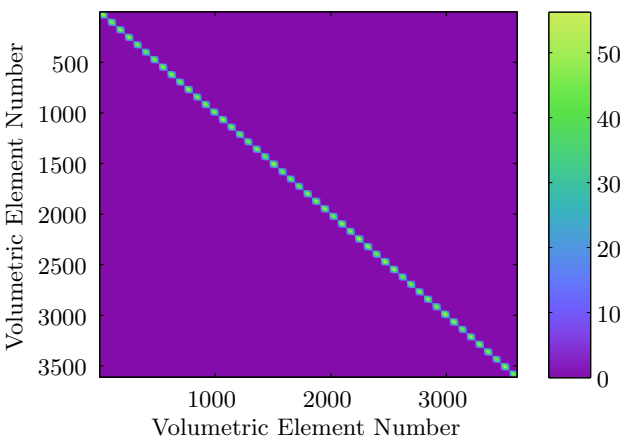


Figure 4.11: 3D a priori model covariance estimate

4.5 Simulated Inversion Results

This section will discuss the results of the inversion under the conditions previously described. Three visualization methods will be employed to assess the accuracy of the solution. In the first, several blocks will be selected and a depth-dependent comparison will be made, similar to what was done with the 1D examples.

Figure 4.12 shows which blocks will be explored. The magenta boxes in the figure correspond to the depth-dependent results in Figure 4.13. The identifying letters in Figure 4.12 correspond to the identifiers in the titles of Figure 4.13. The specific magenta blocks were selected because the region was created to vary smoothly (but not necessarily linearly) from the profile in the block $(1, 1)$ to the profile in block $(7, 7)$. The results in Figure 4.13 are representative of the transition in the region.

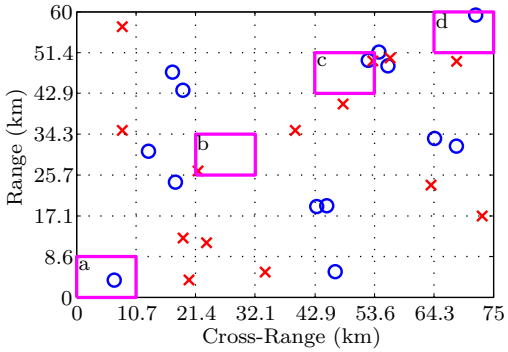


Figure 4.12: Illustrations of selected range-independent blocks

Figure 4.13(a) has the same sound-speed profile as the 1D examples have. The accuracy and confidence intervals of the 3D inversion in this block are on the same order as those seen in the 1D inversion. These results show one of two extremes in the region. Figure 4.13(d) shows the other extreme in the region. This block has the simplest sound-speed profile. As expected, the inversion accuracy is improved for the simpler sound-speed profile. The

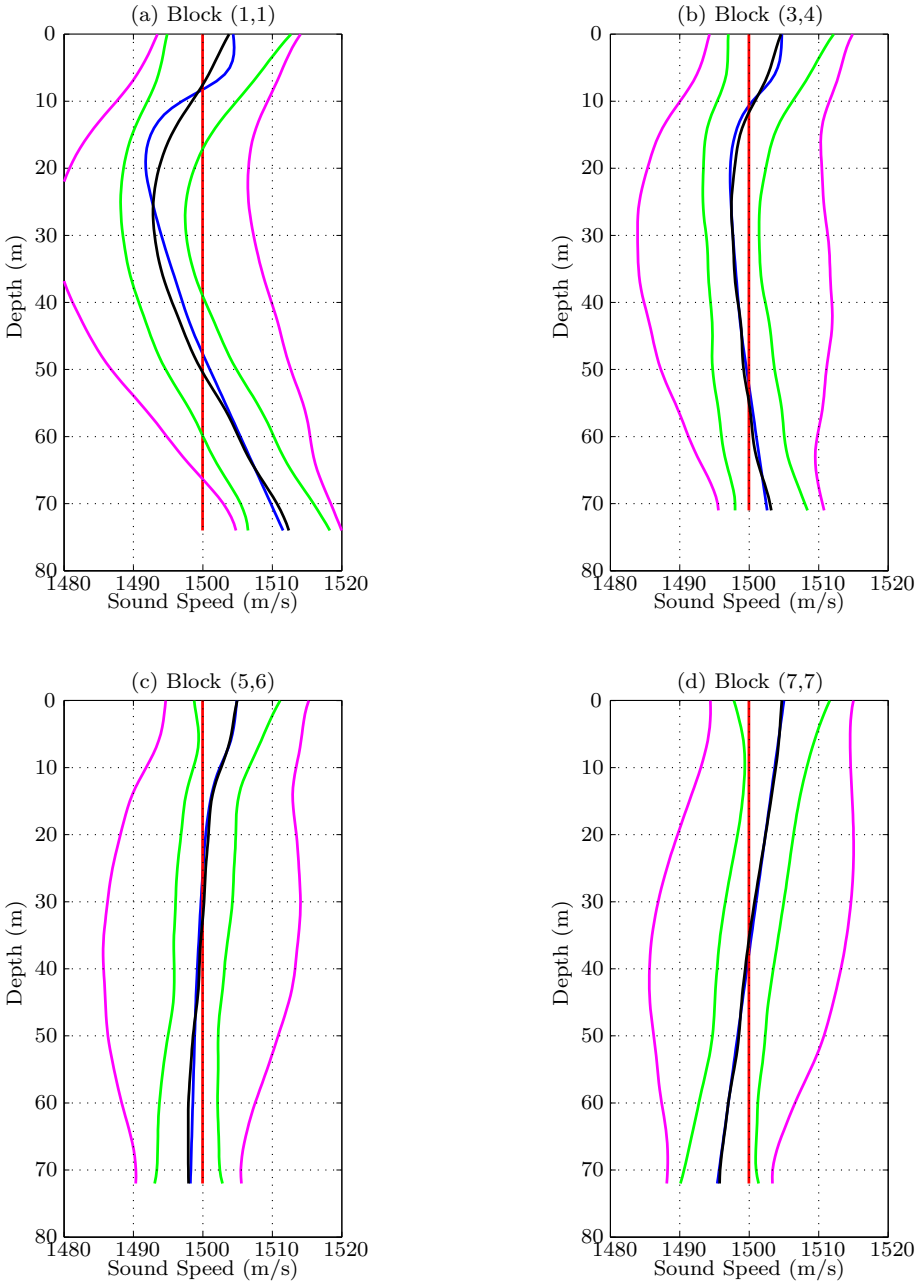


Figure 4.13: Range-independent 3D inversion results for the blocks shown in Figure 4.12, a simulated environment (See Figure 4.3 for legend)

improvement in overall accuracy for simpler sound-speed profiles is a consequence of the true solution conforming well to the solutions preferred by the constraints. Figures 4.13(b) and (c) show increasing accuracy with decreasing profile complexity. All the confidence intervals in Figure 4.13 are equitable in width. This is a consequence of the fact that the part of the *a priori* covariance matrix for each block is equitable. Additionally, the *a priori* data covariance matrix is invariant over the region and the part of the kernel that corresponds to each block is very similar since the profiles in each block are all on the same order.

The second visualization method examines the accuracy of the solution of each block of the search area. The accuracy is assessed by taking the difference between the inverted profile and true profile. The difference is then averaged in depth and the result is plotted. This relationship is given mathematically as

$$\epsilon(x,y) = \frac{1}{D_{x,y}} \sum_{d=1}^{D_{x,y}} \left(c_{\text{inv}}(x,y,d) - c_{\text{true}}(x,y,d) \right) \tag{4.3}$$

where ϵ is the error in a particular block, $D_{x.y}$ is the maximum depth in block (x,y), d is the depth element, c_{inv} is the inverted sound speed, and c_{true} is the actual sound speed. This result is illustrated in Figure 4.14.

An examination of Figure 4.14 shows that overall, the error is low. Sources and receivers have been included in the figure to illustrate the locations of the acoustic paths. The average error in the majority of the blocks is near

zero. This method allows for observation of the direction of the errors, but also allows for errors to cancel out in each block as mentioned earlier. Examination of the error in Figure 4.14 shows that blocks with no acoustic data actually out performs some of the other blocks. In this case, the background profile was fairly representative of the actual profile.

Blocks $(1,2)$ and $(1,3)$ have the highest average errors in the region. By examining where the acoustic paths lie, it is apparent that only two acoustic paths travel through those blocks. Since those paths pass through other blocks as well, there is some ambiguity concerning the results in those blocks leading to an over estimation in one and an underestimation in another.

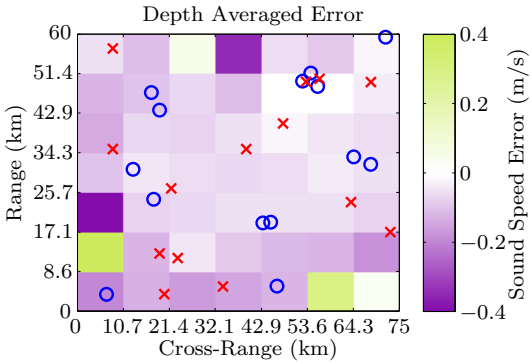


Figure 4.14: Error between inverted and actual sound speeds averaged over depth

The final visualization method is very similar to the second method. Instead of examining the depth averaged error in the region, it examines the L2 error between the inverted and actual sound speeds, averaged over depth. This relationship is given mathematically as

$$\epsilon_{L2}(x,y) = \left(\frac{1}{D_{x,y}} \sum_{d=1}^{D_{x,y}} \left(c_{\text{inv}}(x,y,d) - c_{\text{true}}(x,y,d) \right)^2 \right)^{1/2} \tag{4.4}$$

where ϵ_{L2} is the depth averaged L2 error. Figure 4.15 illustrates the depth averaged L2 error for the region. This result also shows that the inversion did very well. An inspection of where the inversion performs best shows that the location contains the greatest density of acoustic paths. Blocks with fewer acoustic paths do not perform as well.

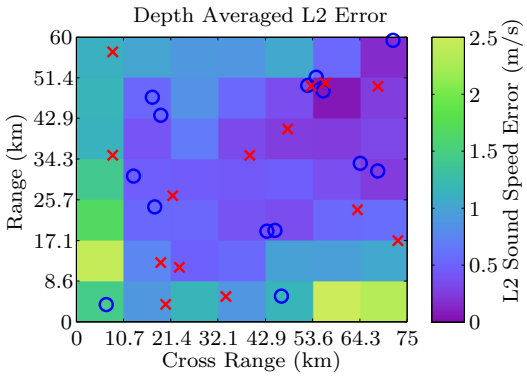


Figure 4.15: L2 error between inverted and actual sound speeds averaged over depth

4.6 Hudson Shelf Region

The rest of this chapter will focus on information provided in an oceanographic model of the Hudson Shelf off the coast of New Jersey, USA. This model provides a high resolution approximation of real-world conditions in the region and was created by researchers at the Woods Hole Oceanographic Institute [29]. It is convenient to continue to use simulations because it is possible to make a direct comparison of the inversion to the actual sound speeds. The model does not provide any sound speed information. However, the information in the model (depth, temperature, and salinity) can be used to determine sound speed. This work will utilize the Nine-term Mackenzie Equation [31].

Figure 4.16 shows the surface temperature in the region to put into context the level of variability in the region. The figure also shows the latitude and longitude coordinates in which the region is located. The region is too vast for the scope of the present work. Instead, this work will focus on the area outlined by the black box which consists of 91×91 range-independent blocks. This particular spot was chosen because the sound-speed field is complex, but the bathymetry varies slowly in range and cross-range. This region is recast into a Cartesian coordinate system from latitude and longitude coordinates.

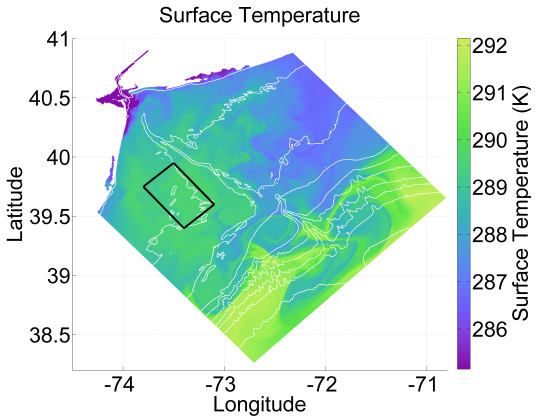


Figure 4.16: Calculated surface temperature on the Hudson Shelf based on oceanographic modeling

4.6.1 Differing Generation Methods

One of the principle questions that needs to be answered about 3D inversion is how well an inversion represents a region. The previous examples have shown that the inversion can recover a sound-speed profile when the data is generated on the same grid as the inversion is performed. In this case, the true sound speeds have no variation within each block. In reality, the sound speed in the ocean must vary smoothly in all three dimensions. Prior to this work, no information was available to ascertain how the 3D inversion

will perform in a continuously varying ocean.

This section will attempt to answer that question by performing a case study on how the data is generated. The first case will be referred to as the coarse case and it utilizes the same method as in Section 4.5 where the data is generated on the same gridding as the inversion. In this case, the oceanographic model is divided into a 7×7 grid. Each block in this 7×7 grid is then composed of a 13×13 grid. One block in the 13×13 is then chosen to represent the others. Equation (4.1) is then applied on this 7×7 grid and the inversion is performed on the resulting travel times. This case is essentially the control case as it has already been shown how the inversion performs under these circumstances.

The second data generation case will be referred to as the smooth case. In this case, the water depths and sound speeds have been artificially modified so that the changes in both parameters vary slowly with range and cross-range. This smoothing operation for the sound speed is given as

$$c_{\text{smooth}}(x, y, z) = c_{\text{true}}(x, y, z) * *W(x, y) \tag{4.5}$$

where c_{smooth} is the smoothed sound-speed field used in data generation, c_{true} is the sound-speed field found in the oceanographic model, W is a window function, and the $**$ represents convolution in x and y. The window function was a scaled rectangular distribution in range and cross-range. Mathematically, the window function is

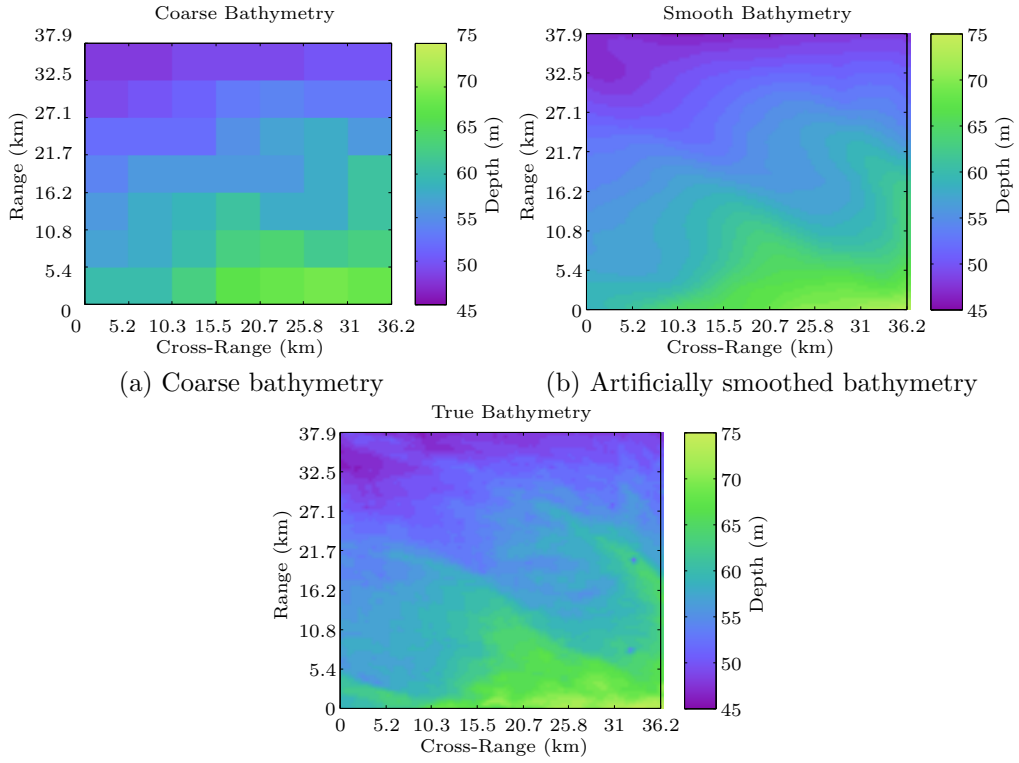
$$W(x,y) = \frac{1}{169}\mathrm{rect}\left(\frac{x}{13\Delta x}\right)\mathrm{rect}\left(\frac{y}{13\Delta y}\right) \tag{4.6}$$

where Δx is the width of each block in cross-range and Δy is the length in range. The factor of 13 is the number of blocks that was used to locally smooth. The same procedure was performed on the bathymetry. Equation (4.1) was applied to the result (which was still a 91×91 grid). The inversion was then performed on the resulting travel times, but on the same 7×7 grid as in the coarse case. This allows for easy comparison of performance between the cases.

The third and final data generation case will be referred to as the true case. In this case, the exact data available from the oceanographic model are used. This case should be most representative of how an actual experiment in the region would perform under the same inverted gridding.

In addition to the data generation case study, another case study was performed simultaneously. In this study, the effects of spatial sampling resolution were explored. All the examples have used no less than the minimum number of acoustic paths needed to properly define the solution (though the placement of these acoustic sources has not always been appropriate). This study explored the effects of under-defining the problem, referred to as the low case; defining the problem (in ideal configurations, which were not utilized), referred to as the moderate case; and over-defining the problem, referred to as the high case.

As a result, there were nine different cases investigated. Figure 4.17 shows the bathymetry for the different data generation cases. This helps to illustrate the differences in the generation methods. In Figure 4.17(a) the resolution is very low, in Figure 4.17(b) the resolution is higher but shows very slow changes, and in Figure 4.17(c) the resolution is higher and the changes more erratic. Looking from (c) to (a) in Figure 4.17, the reduction in resolution from the truth is apparent. The ability of the inversion to capture the various levels of variability is precisely the question this section address.

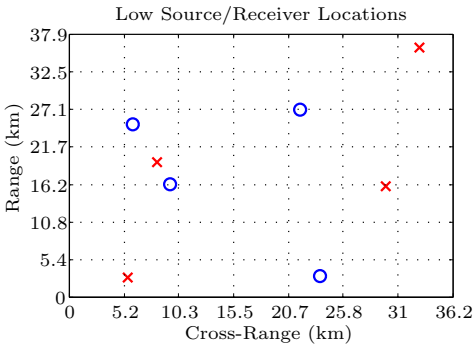


(a) Coarse bathymetry

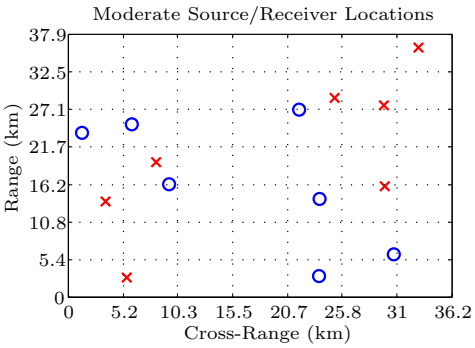
(b) Artificially smoothed bathymetry

(c) True bathymetry

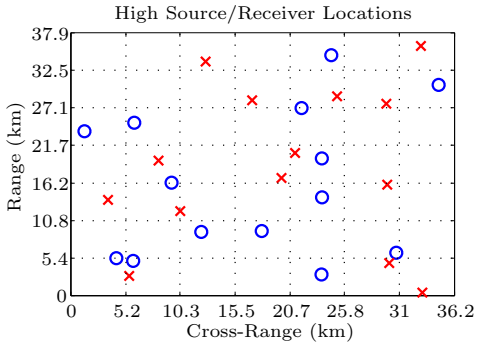
Figure 4.17: Illustrations of bathymetry for the three differing data generation methods



(a) Low spatial sampling source/receiver locations



(b) Moderate spatial sampling source/receiver locations

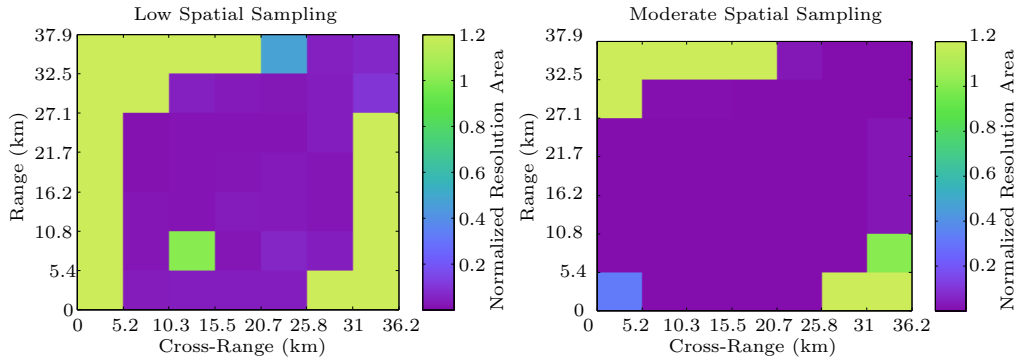


(c) High spatial sampling source/receiver locations

Figure 4.18: Source and receiver locations for various levels of spatial sampling

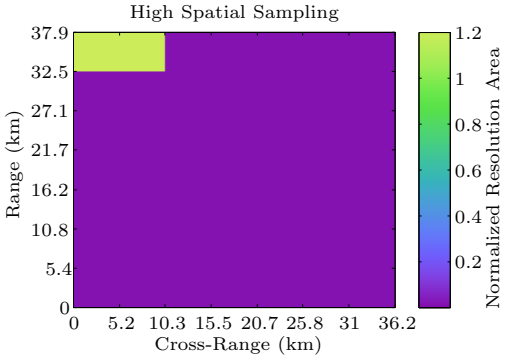
Figure 4.18 shows the locations of the sources and receivers for the different levels of spatial sampling. Figure 4.18(a) shows the location of the four sources and four receivers on the inversion grid, Figure 4.18(b) shows the location of the seven sources and seven receivers, and Figure 4.18(c) shows the location of the fourteen sources and fourteen receivers. The locations of the sources and receivers does not change from case to case. The higher levels of spatial sampling add sources and receivers without moving the transducers

from the sparser cases. In all of these cases, the experimental setup uses the acoustic swarm as was used in the previous 3D example.



(a) Low spatial sampling 2D resolution

(b) Moderate spatial sampling 2D resolution



(c) High spatial sampling source/receiver locations

Figure 4.19: 2D resolution (in range and cross range) for various levels of spatial sampling

Figure 4.19 shows the 2D resolution length of Eq. (3.45) for the spatial sampling cases. Low resolution length means the solution is well resolved, while high resolution lengths mean the solution is poorly resolved.

Unlike the resolution defined in Eq. (3.45) the resolution plotted in

Figure 4.19 has been normalized such that the maximum finite resolution length in Figure 4.19(a) is unity. The same normalizing constant is used in both Figures 4.19(b) and (c) so that they will be on the same scale. This normalization is not detrimental because the primary use of the resolution length is relative—the absolute value of the resolution length in a particular block is immaterial as the scale of each inversion depends on the range and cross-range discretization.

Figure 4.19(a) shows the resolution for the low spatial sampling case. This case has the greatest amount of variation in resolution length because the sampling is so low. This case also has blocks that are not uniquely defined. This occurs because the number of acoustic paths is so low that a single path is the only path that runs through multiple blocks resulting in an ambiguity. Figure 4.19(b) has some variation, but the majority of the region has gone to zero because the acoustic paths have over-defined those blocks. Figure 4.19(c) has zero variation because the spatial sampling is great enough that the vast majority of the region is over-defined. An obvious feature across each of the cases is the blocks with a resolution length greater than unity. This would seem impossible based on the method of normalization. However, these blocks have no acoustic paths and therefore the resolution length in these blocks in infinite. As expected, increasing the spatial sampling reduces the number of unresolved blocks and decreases the resolution of the sampled blocks.

4.6.2 Hudson Shelf Inversion Results

As in Section 4.5, this section will show the results from the case studies through three visualization methods. Unlike the visualization in Section 4.5, this section will only show one range-independent inversion result for each case but will utilize the same block for each case. Specifically, block $(3, 3)$ is used for the depth-dependent results. The expected result in this analysis is that accuracy will increase and uncertainty will decrease for increasing spatial sampling. Additionally, it is expected that accuracy will decrease and uncertainty will increase with increasing environmental variability.

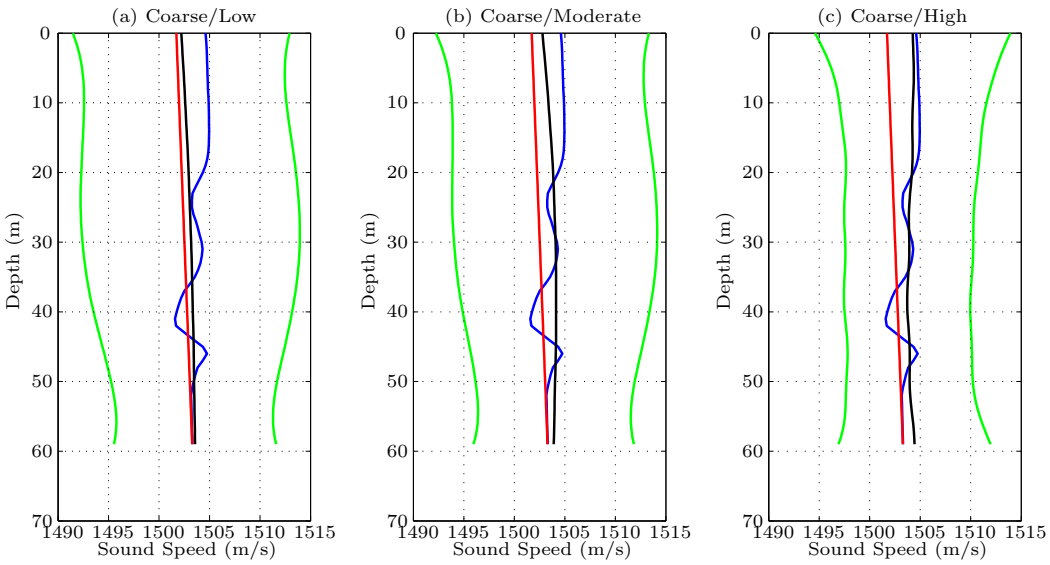


Figure 4.20: Range-independent 3D inversion results in depth for block $(3, 3)$ using coarse generation and (a) low, (b) moderate, and (c) high spatial sampling (See Figure 4.3 for legend)

Figure 4.20 shows the depth-dependent results in block $(3, 3)$ for each of the spatial sampling cases in the coarse generation case. As in the previous

sections, the green lines represent the 95% confidence interval, the blue line represents the actual sound-speed profile, the black line represents the inversion result, and the red line represents the background profile. In these case studies, the background profile was determined by performing a first-order fit to an assumed measurement somewhere in the search area. The same background profile is used across all studies and is a first-order fit instead of a zero-order fit as in the previous examples. These results no longer show the *a priori* confidence intervals since the focus of these results is on comparing case studies and not on the individual improvements.

Figure 4.20(a) corresponds to the low spatial sampling case. The figure shows that the inversion has started to move towards the actual profile and have a fairly wide confidence interval. Figure 4.20(b) corresponds to the moderate spatial sampling case and shows improved accuracy and slightly decreased confidence interval width. Figure 4.20(c) corresponds to the high spatial sampling case and shows considerably improved inversion results and considerably narrower confidence intervals.

Figure 4.21 shows the depth-dependent results in block $(3, 3)$ for each of the spatial sampling cases in the smooth generation case. Before beginning it is necessary to note the considerable increase in the number of actual profiles in the figures. This is because the actual sound speed in each range-independent result is composed of the 13×13 range-independent smoothed profiles.

Figure 4.21(a) corresponds to the low spatial sampling case and shows wide confidence intervals and an inversion solution that has moved towards

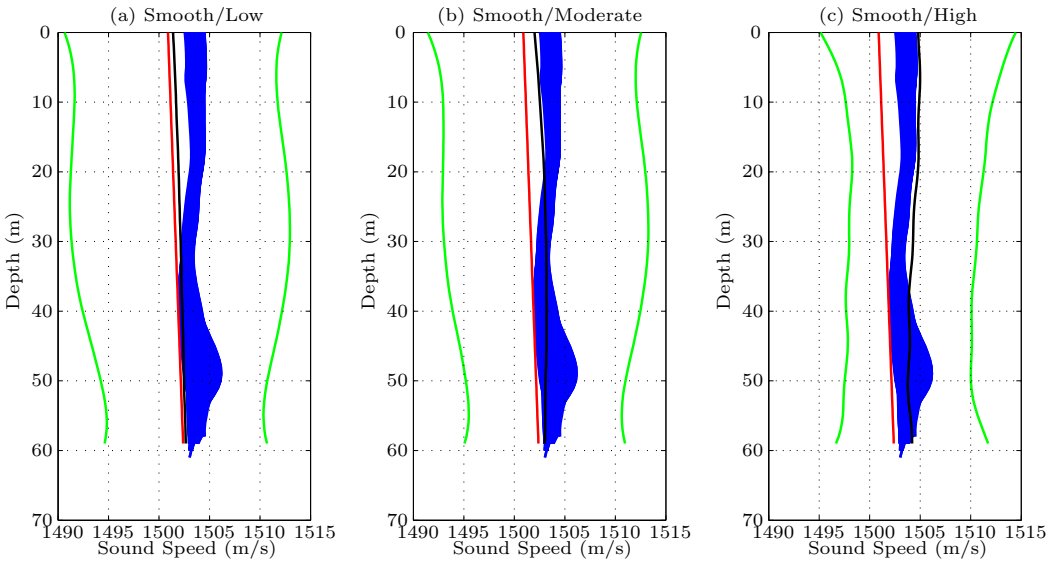


Figure 4.21: Range-independent 3D inversion results in depth for block $(3, 3)$ using smooth generation and (a) low, (b) moderate, and (c) high spatial sampling (See Figure 4.3 for legend)

the actual sound-speed distribution. Figure 4.21(b) corresponds to the moderate spatial sampling case and shows little change in confidence interval width but show considerable improvement in inversion accuracy. Figure 4.21(c) corresponds to the high spatial sampling case and shows considerably narrower confidence intervals, but the inversion seems to have moved past the actual sound-speed distribution.

It is difficult to say whether this result is worse as there is no longer an actual sound-speed profile for the block. However, it will be assumed that the inversion result should be contained within the actual sound-speed profile distribution in each block. With this assumption in mind, it would appear the high spatial sampling case has over-corrected. This is a possible result as the

additional data may be over-influenced by the surrounding blocks such that overall accuracy in the region is improved at the cost of decreased accuracy in particular blocks.

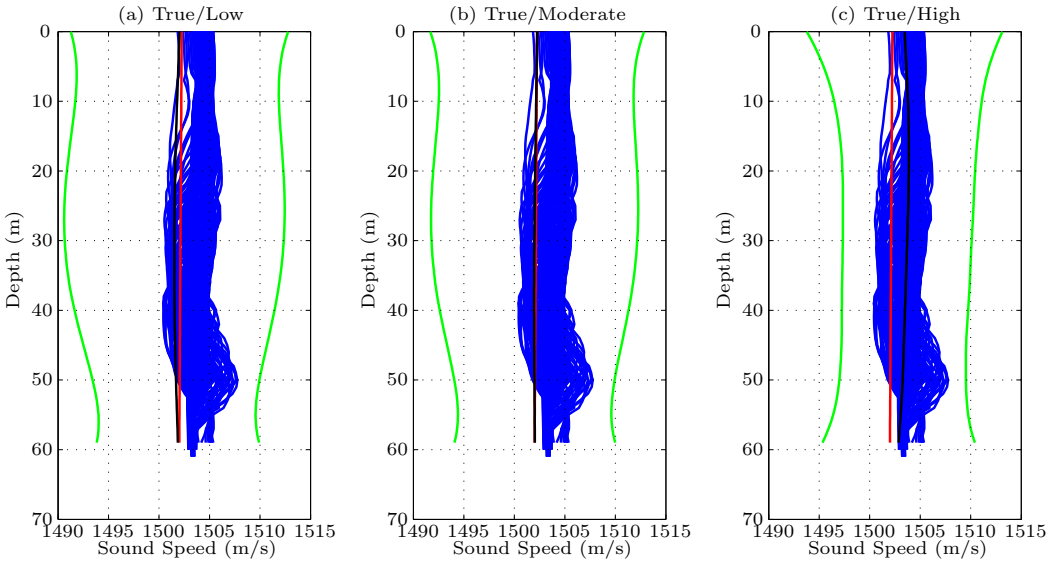


Figure 4.22: Range-independent 3D inversion results in depth for block $(3, 3)$ using true generation and (a) low, (b) moderate, and (c) high spatial sampling (See Figure 4.3 for legend)

Figure 4.22 shows the depth-dependent results in block $(3, 3)$ for each of the spatial sampling cases in the true generation case. In this figure, the actual sound speeds show considerably more variability than in the smooth case. This is because the true generation case was not modified to assure the range and cross-range variability remains low.

Figure 4.22(a) corresponds to the low spatial sampling case and shows wide confidence intervals and an unexpected inversion result. In this case, the inversion result has moved away from the center of the actual sound-speed pro-

file distribution but is still contained within the extremes of the distribution. Figure 4.22(b) corresponds to the moderate spatial sampling case and shows equitably wide confidence intervals to the previous case and an inversion result that has moved back towards the center of the distribution. The result of this move towards the center of the distribution means that the inverted profile is very nearly the background profile. Figure 4.22(c) corresponds to the high spatial sampling case and shows notably narrower confidence intervals and an inversion result that appears to be representative of the sound-speed profile distribution.

A comparison between Figures 4.20(a), 4.21(a), and 4.22(a) shows little to no change in confidence interval width and a nominal loss of accuracy as variability is increased. Making the same comparisons across the data generation cases for the other other levels of spatial sampling shows the same trend.

As before, the second visualization method takes the depth averaged error between inversion and truth. This is somewhat more complicated in the smooth and true data generation cases where there is no longer a single true profile. To attempt to solve this problem the following modification is made to Eq. (4.3):

$$\epsilon(x,y) = \frac{1}{D_{x,y}} \sum_{d=1}^{D_{x,y}} \left(c_{inv}(x,y,d) - \frac{1}{XY} \sum_{y_0=1}^{Y} \sum_{x_0=1}^{X} c_{true}(x_0,y_0,d) \right) \tag{4.7}$$

where Y and X are the number of range-independent blocks in range and

cross-range within each inversion block and y_0 and x_0 are the identifiers of said sub-blocks. Figure 4.23 shows the depth averaged error ϵ for each of the nine cases. Equation (4.7) assumes the mean true profile is most representative of the actual profiles in each block.

There are two primary trends to observe in the figure. Progressing from left to right corresponds to increasing spatial sampling and shows a decrease in the average error. Progressing from top to bottom corresponds to increasing the environmental variability and shows an increase in the average error. Both of these trends are in agreement with expected results and with the results from the single-block results in Figures 4.20, 4.21, and 4.22. Additionally, a comparison between the results in Figure 4.23 and the resolution area in Figure 4.19 helps to illustrate the usefulness of the resolution area. This comparison shows agreement in the location where the error is highest as it corresponds to a location with zero acoustic paths. Comparisons of other blocks with infinite resolution area shows where the background profile happens to be representative of the truth.

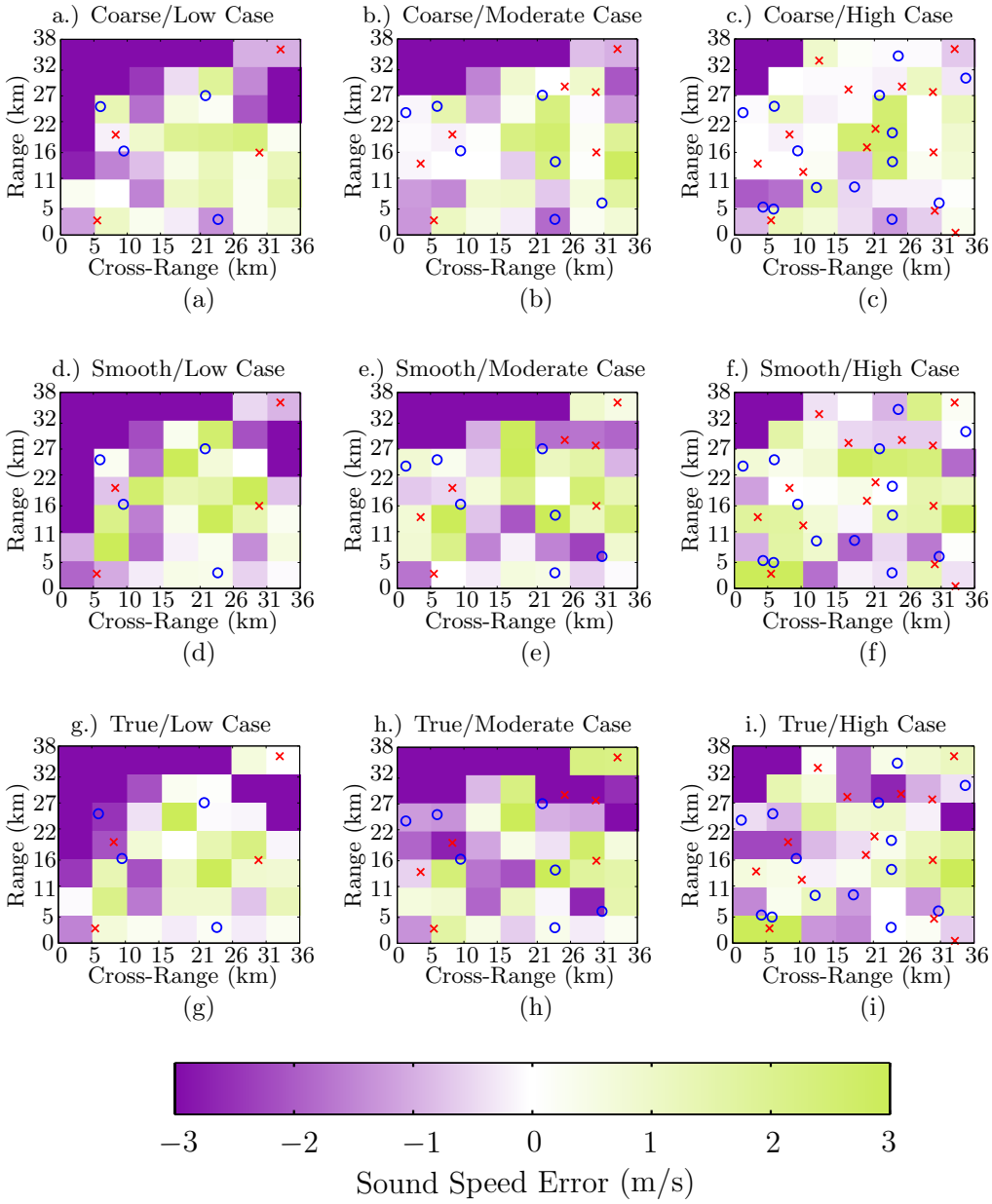


Figure 4.23: Depth averaged error for various cases

The third visualization method takes the depth averaged L2 error between inversion and truth. As with the depth averaged error, a modification is necessary to handle the smooth and true cases. Equation (4.4) becomes

$$\epsilon(x,y) = \left(\frac{1}{D_{x,y}} \sum_{d=1}^{D_{x,y}} \left(c_{\text{inv}}(x,y,d) - \frac{1}{XY} \sum_{y_0=1}^{Y} \sum_{x_0=1}^{X} c_{\text{true}}(x_0,y_0,d) \right)^2 \right)^{1/2}. \quad (4.8)$$

The results of this analysis are given in Figure 4.24. Looking from left to right shows a decrease in the average L2 error corresponding to the increase in spatial sampling. Looking from top to bottom shows an increase in the average L2 error corresponding to the increase in environmental variability. Both of these trends are in agreement with the expected results. Additionally, the trends are similar to those shown by other analytic tools such as average error and resolution.

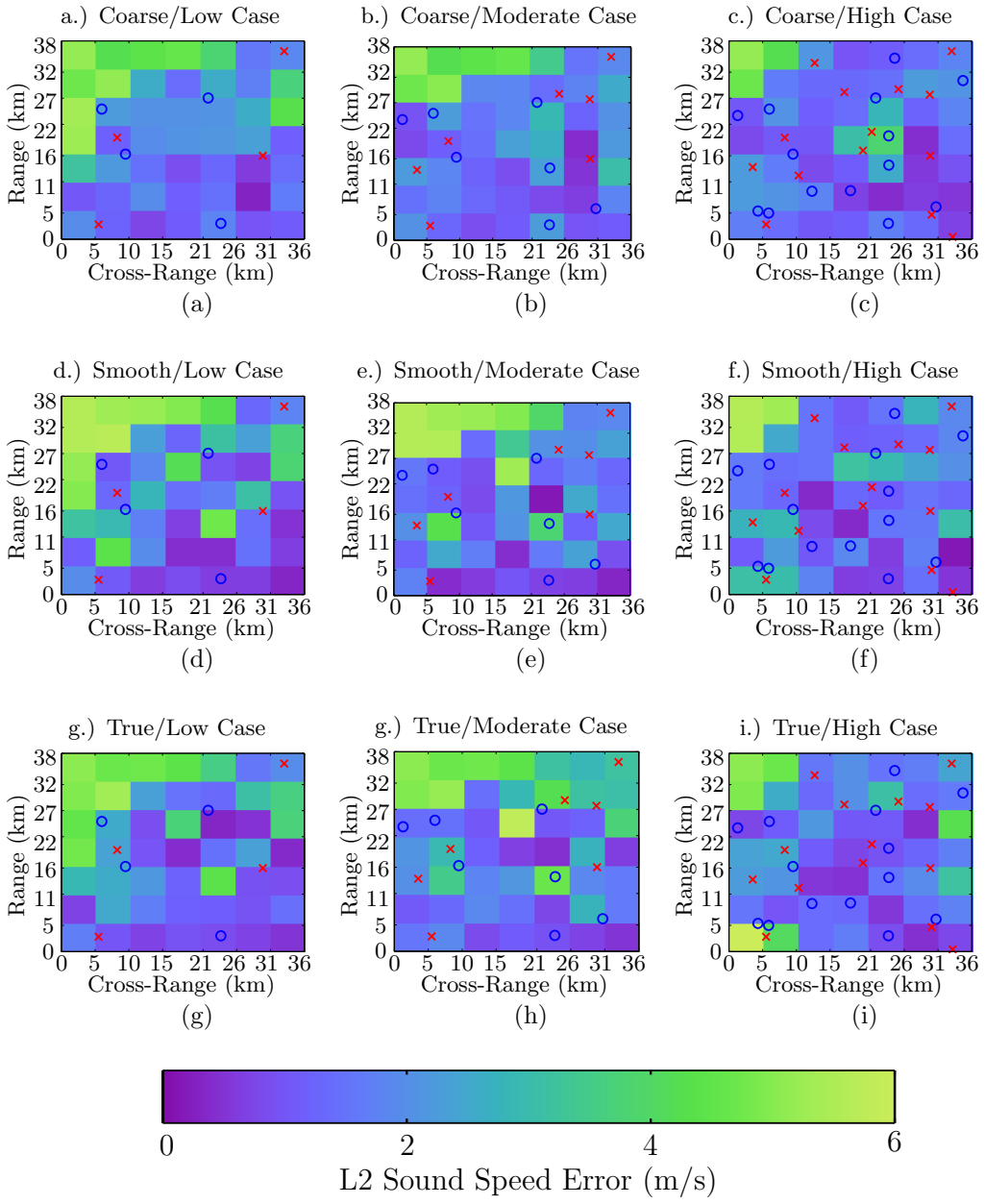


Figure 4.24: Depth averaged L2 error for various cases

Unlike in Section 4.5 two additional visualization methods will be discussed. The first gives a single average L2 error value for each case and is shown in Figure 4.25. This result is convenient because it gives a single number evaluation of the each case study and makes comparison more straight forward. The results in Figure 4.25 shows that generally increasing spatial sampling decreases error and that increasing environmental variability increases error. These trends are not followed by the true generation low spatial sampling case. This case denies both trends. However, an examination of the acoustic travel time data can explain the discrepancy.

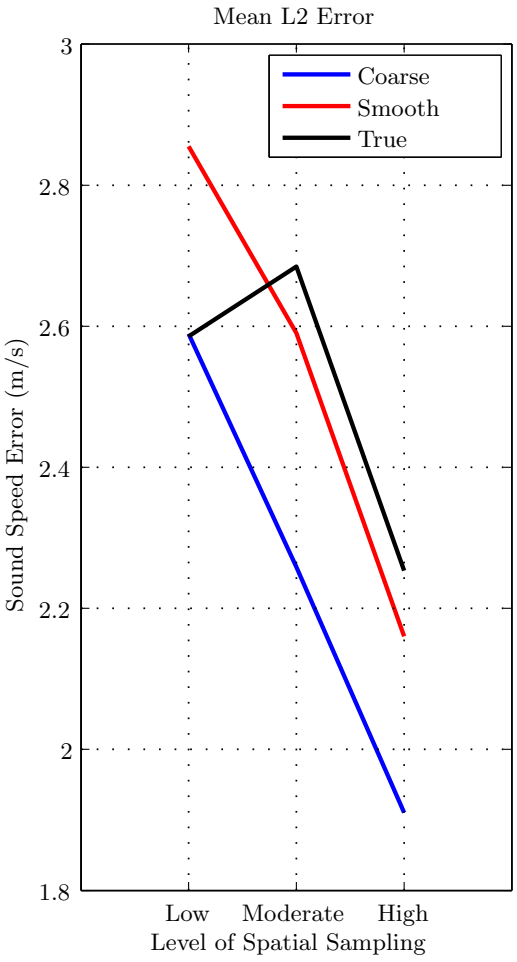


Figure 4.25: Mean L2 error across entire region for each case

The summation of the travel times for each case are given in an adjusted format in Figure 4.26. The adjustment is made by subtracting the travel time of the coarse case from each case. This subtraction helps to put all the travel time data on the same scale so that comparisons are more convenient. Under the assumption that the coarse cases will perform best, than the nearer the

data for the other cases match the data for the coarse case, the better their performance. Figure 4.26 only shows the summation of the travel times and not each data point, however, it should be a good first-order approximation of how closely the data compare.

Since the true generation low spatial sampling travel time summation is almost the same as the coarse generation low spatial sampling travel time summation it makes sense that the results would be similar. Essentially what this means is that the unusually high accuracy of the true/low case is an artifact of unintentionally convenient spatial under-sampling.

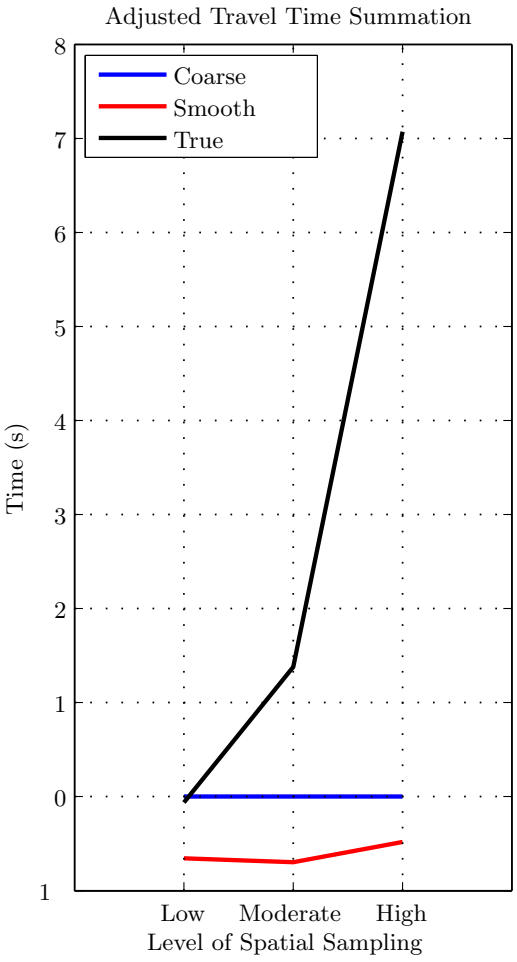


Figure 4.26: Adjusted summation of inversion data

The final visualization method is given in Figure 4.27. The information in this figure shows the *a priori* and *posteriori* standard deviations in each case. This is the only figure that only shows information concerning the uncertainty of the results.

Figure 4.27 shows that the general confidence interval trends seen in the single range-independent blocks of Figures 4.20, 4.21, and 4.22 holds, on average, for the region. Figure 4.27 shows that increasing spatial sampling decreases the *posteriori* standard deviation. However, environmental variability has little to no effect on the *posteriori* standard deviation. The emboldened black line indicating the *a priori* standard deviation has been included to emphasize that each case utilized the same starting assumption and to show how the inversion has improved from the *a priori* model covariance estimate.

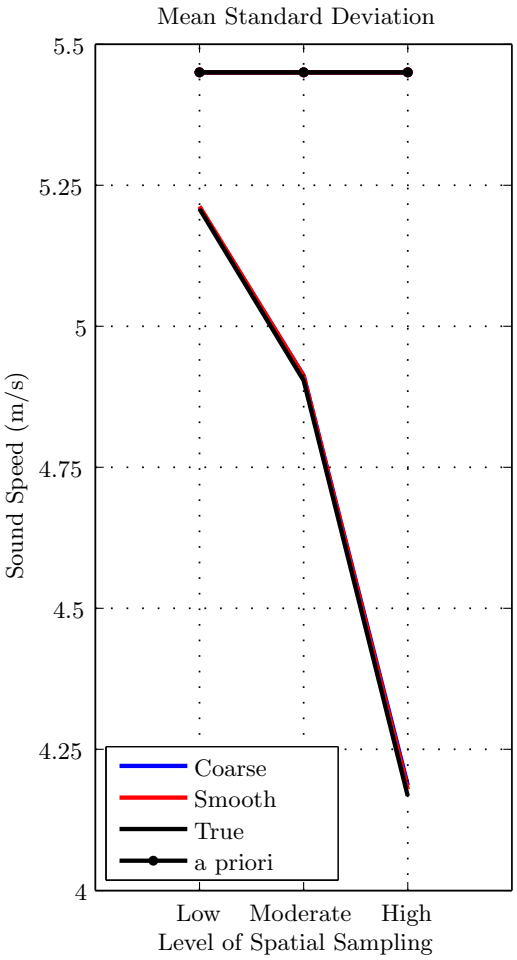


Figure 4.27: *a priori* and *posteriori* mean standard deviation

The only deviation from expected trends concerns the uncertainty of the inversion with respect to environmental variability. It was expected that increasing environmental variability would lead to an increase in the uncertainty but the results show otherwise. A second look at Eq. (3.30) shows that there are three distinct parameters that contribute to the inversion covariance. One of these parameters, the *a priori* model covariance, has been chosen such

that it is the same for all the cases. This choice was made because, in application, there would not be a reasonable way to vary this parameter with environmental variability since the variability would not be known. This leaves two other parameters to explore.

The *a priori* data covariance matrix was generated by assuming a standard deviation of 1% of the travel time measurement. Since the deviation of the summation of several hundred travel times was on the order of ten seconds (illustrated in Figure 4.26), the variation in the *a priori* data covariance matrices should also be extremely small. The final parameter is the inversion kernel. The kernel is largely composed of the estimated environmental information. Since the environments of each case are very similar it stands to reason that the kernels should also be similar.

4.7 Summary

In this chapter several simulated examples of inversions were investigated. The first example was in 1D. In this case, the inversion performed adequately with a single acoustic path. The addition of acoustic paths was shown to improve the 1D accuracy and decrease uncertainty. Several forms of error were investigated for the 1D case. It was shown that realistic levels of uncertainty in source/receiver locations the would not introduce significant error into the inversion result. The addition of random noise to the data could introduce errors in the inversion if the travel time errors were significantly large. This failure was countered by adjusting the *a priori* data covariance matrix

but at the cost of decreasing the solution's depth-dependent resolution. Finally, errors in bathymetric estimations were shown to introduce significant errors in the 1D inversion result.

Next, a simulated 3D region was introduced and an inversion was performed. Three data visualization methods were utilized to investigate the results. The depth-dependent results showed high accuracy and narrow confidence intervals. The depth averaged error also showed high accuracy where acoustic sampling was high and fair results where sampling was lower. Similarly, the depth averaged L2 error showed equitable levels of accuracy.

The final sections introduced an oceanographic model of the Hudson Shelf region. This high resolution oceanographic model was used to perform several simulated case studies in a realistic environment for varying levels of spatial variability and spatial sampling. The results from this case study showed that increasing spatial sampling increases inversion accuracy and reduces inversion uncertainty. Additionally, increasing spatial variability decreases inversion accuracy and has no effect on inversion uncertainty.

Chapter 5

Conclusions

5.1 Summary

This work presented a 3D perturbative inversion scheme for obtaining estimates of sound-speed fields in the shallow ocean. In Chapter 3, two methods for constraining the problem to improve stability and establish uniqueness were introduced. The two methods were combined to produce a hybrid solution. Then several inversions were performed on simulated data to assess the functionality of the inversion scheme.

The first stabilization method was referred to as the method of approximate equality constraints and utilized a combination of second order Tikhonov regularization and an absolute equality constraint. The second stabilization method was referred to as the covariance method and utilizes *a priori* estimates of the data and model covariance matrices. The hybrid method used Tikhonov regularization and the absolute constraint in the construction of the *a priori* model covariance matrix. The combination of these two methods allows for specification of solution characteristics as well as knowledge of the model and data covariances. The solution also allows for adjustment of the relative strengths of the different characteristics.

In Chapter 4, the solution mechanism was applied to several examples. The first example was an inversion in 1D with the exact number of acoustic paths needed to define the environment: one. The result from this inversion showed high accuracy between the inversion result and the true sound-speed profile. The inversion also showed a significant decrease in the width of the confidence interval from the *a priori* estimate. Next, the inversion was performed on the same example with the addition of acoustic paths. The increased sampling showed a marked increase in the accuracy of the solution in the center of the waveguide but minimal increase near the sea surface. This limitation on improving accuracy of the inversion at the sea surface is a weakness of this technique that can not be improved upon.

The use of the 1D example was utilized to show the effects of various types of error on the inversion solution. The inversion proved itself to be robust well beyond the point at which errors in source/receiver locations would be present— a large 0.1% error in path length resulted in a 2% increase in RMS error from the control example. The inversion demonstrated it could handle moderate amounts of random noise on the data, but failed if the noise grew beyond a certain point. This was shown when the travel time data error was 1.0% and the RMS error increased by 943%. Modification of the *a priori* data covariance is able to counter the instability of increased noise (RMS error increased by 233% instead of 943%), but at the cost of allowing the *a priori* information to have a greater effect on solution resolution. The inversion in 1D showed that errors in bathymetry could lead to considerable errors in the

inversion result; a 1 m inaccuracy in depth yielded a 140% increase in the RMS error.

Next, a full 3D environment was generated, an abundance of acoustic paths implemented, and the inversion technique applied. The results from the inversion in the simulated 3D region proved to have high accuracy, with an RMS difference of 0.98 m/s between the inverted and actual sound speed, and low uncertainty, with an average standard deviation of 5.76 m/s. To test the inversion against a more realistic inversion, an oceanographic model of the Hudson Shelf was used. This model allowed for a more realistic environment because the resolution of the information in range and cross-range was high enough to approximate the continuous nature of the real ocean. This resolution presented the opportunity to see how well the inversion performed when the range and cross-range discretization of the inversion is considerably lower than the rate at which the sound speeds change.

To assess the effects of environmental variability, a case study was performed in which the data was generated on the same gridding as the inversion, when the data gridding was much more resolved but the variability was still low, and when the data gridding and variability were high. Additionally, a case study was performed to test how well the inversion performed under different levels of spatial sampling. The results from the simulated study showed that the inversion accuracy increased and the uncertainty decreased with increased spatial sampling. Additionally, the inversion accuracy decreased with increased spatial variability; however, the inversion uncertainty was unaffected

by the spatial variability.

5.2 Future Work

It was shown how the inversion performs for an assumed inversion grid for differing levels of spatial sampling and variability. In application, it would be helpful to be able to select the inversion grid for an expected level of variability. Then a determination of the number of acoustic transducers could be made. These are all necessary steps in designing a real-world experiment. So far, little information on the appropriate inversion gridding is available. Additionally, a determination of the best orientation of the inversion gridding could be helpful.

An investigation into the effect horizontal refraction has on the inversion could help to provide insight into potential complications. Horizontal refraction could introduce errors in path length similar to what was discussed in Section 4.2.2.1. However, horizontal refraction could introduce sampling errors since the inversion assumes straight acoustic paths. In this case, the technique would attribute the group speeds along on the refracted path to the sound-speed correction in the blocks along the straight path.

This work focused on proof of concept and determining the best operating conditions for the 3D inversion technique, but the real application is in the real-world shallow ocean. The use of real data is a necessary step in the development of this inversion technique.

Appendix

Matlab Algorithms

This appendix contains several of the Matlab scripts necessary to perform an example inversion. It does not contain any of the necessary scripts to run Kraken. Information on how to use Kraken can be found at [30].

Setup Algorithm

```
% This script generates the necessary data to perform an inversion
clear all; close all; clc

% Generate Profile
cl = [1540.00 1540.00 1540.00 1540.00 1540.00 1540.00 1539.00 1537.00 ...
      1534.00 1529.00 1518.00 1507.00 1500.00 1494.00 1490.00 1487.00 ...
      1484.00 1482.50 1481.50 1480.25 1480.00 1480.00 1480.50 1481.00 ...
      1481.50 1482.10 1482.80 1483.50 1484.20 1484.90 1485.80 1486.70 ...
      1487.60 1488.50 1489.40 1490.30 1491.20 1492.10 1493.00 1493.90 ...
      1495.00 1496.30 1497.70 1499.15 1500.65 1502.15 1503.70 1505.20 ...
      1506.70 1508.20 1509.70 1511.20 1512.70 1514.20 1515.70 1517.20 ...
      1518.70 1520.20 1521.70 1523.20 1524.70 1526.20 1527.70 1529.20 ...
      1530.70 1532.20 1533.70 1535.20 1536.70 1538.20 1539.70]';

% Generate Number of Sources
% Generate source and receiver locations

% Generate Number of Sources
NoS = 1;

% Generate Number of Receivers
NoR = 1;

xrange = 75000;   % cross-range distance (meters)
yrange = 60000;   % range distance (meters)
```

```
nS = 0;      % Gives the number of extra sources
nR = 0;      % Gives the number of extra receivers

% Determine descritization based on the number of sources and receivers
% At present no adaptive or optimal grid design is in use
dx = xrange / ( NoR-nR );    % x descritization (meters)
x  = 0:dx:( xrange-dx );     % left grid line for all x descritizations
dy = yrange / ( NoS-nS );    % y descritization
y  = 0:dy:( yrange-dy );     % bottom grid line for all y descritizations

% All sources and receivers are assumed to be at the same depth

% Generate variable bathymetry
% Assume some DC depth about which each block sees some perturbation
dz = 1;      % depth Discretization

depth0 = 75;

gaussmf = @( x , s , c ) exp( - ( x - c ) .^ 2 ./ ( 2 * s .^ 2 ) );

% Clean up and combine profiles
c1 = interp1( 1:71 , c1 , linspace( 1 , 71 , 500 ) , 'cubic' );
c1 = conv( c1 , gaussmf( -20:20 , 10 , 0 ) / ...
     sum( gaussmf( -20:20 , 10 , 0 ) ) , 'valid' );
c1 = interp1( 1:length( c1 ) , c1 , ...
     linspace( 1 , length( c1 ) , depth0 + 1 ) , 'cubic' , 'extrap' );
c1 = c1 + linspace( 0 , 35 , length( c1 ) );
c1 = ( c1 - min( c1 ) ) * .25;
c1 = c1 - mean( c1 ) + 1500;
c0 = linspace( 1505 , 1495 , length( c1 ) );

xn = x / max( x );
yn = y / max( y );

xn( isnan( xn ) ) = 0;
yn( isnan( yn ) ) = 0;

close all
wssp2 = NaN( length( y ) , length( x ) , depth0 + 1 );
depth = depth0 * ones( size( wssp2( : , : , 1 ) ) );
for yy = 1 : length( y )
    for xx = 1 : length( x )

        wssp2( yy , xx , : ) = ( c0 * xn( xx ) + ...
```

```
                c1 * ( 1 - xn( xx ) ) + c0 * sin( pi / 2 * yn( yy ) ) + ...
                c1 * ( 1 - sin( pi / 2 * yn( yy ) ) ) ) / 2;
        wssp2( yy , xx , depth( yy , xx ) + 2:end ) = NaN;

    end
end

% Reshape and stack the sound speed profiles into 2D
% depth goes down, x and y go across, x for a y, then the next y
wssp3 = reshape( permute( wssp2 , [ 2 1 3 ] ) , ...
        length( x ) * length( y ) , max( depth( : ) ) + 1 )';

% sediment sound speed profile of true environment
% Presently it is assumed that the sediment sound speeds are all the same
% and all start at the end of the water column (poor assumption)
Csedt= [ 1550*ones( 1,16 ) , 1600*ones( 1,16 ) , 1650*ones( 1,16 ) ]';

% Generate the group speed and wave numbers for each block
% the minimum number of modes that propogates in all blocks is taken as the
% maximum number of modes allowed from all blocks, this makes it simple to
% analyze, but it throws data away

frequency = 50:50:150;
Max_NM    = 2 .^ ( 1:3 ) * inf;
Inversion_depth = dz * ( length( Csedt ) - 1 );
Weight_Index    = 1:4;
[ cgrp_true0 , K_true0 , ¬ , ¬ ] = ...
    Mod_Make_Background( frequency  , Max_NM , x , y , dz , depth , ...
                         wssp2 , Csedt , Inversion_depth , Weight_Index );

% Generate source and receiver locations
S = zeros( NoS , 2 );
R = zeros( NoR , 2 );

S( 1:NoS,: ) = [ rand( [ NoS , 1 ] ) * xrange , ...
                 rand( [ NoS , 1 ] ) * yrange ];
R( 1:NoR , : ) = [ rand( [ NoR , 1 ] ) * xrange , ...
                   rand( [ NoR , 1 ] ) * yrange ];

[ ¬ , Loc ] = max( depth( : ) );
Sound_Speed_Measurement = wssp3( : , Loc )';

All_X = ( 0:dx:xrange )/1000;
All_Y = ( 0:dy:yrange )/1000;
```

```
% Get the distances each path travels in each block based on gridding and
% S/R locations
paths       = Make_Paths_better( All_X * 1000 , All_Y * 1000 , S , R );
paths_save  = paths;
Pathsize    = size( paths );
% Reorganize the information so that the path data is store in a 2D matrix
paths       = reshape( permute( paths , [ 2 , 1 , 3 ] ) , ...
              length( x ) * length( y ) , NoS * NoR )';
paths0      = paths;
paths_save0 = paths_save;

Data2 = NaN( [ size( paths , 1 ) , size( K_true0{ end } , 3 ) , ...
               length( frequency ) ] );
for ff = 1 : length( frequency )
    for mm = 1 : size( K_true0{ ff } , 3 )
        group_speed( : , mm , ff ) = reshape( permute( ...
            cgrp_true0{ ff }( : , : , mm ) , [ 2 , 1 ] ) , ...
            length( x ) * length( y ) , 1 );
        Data2( : , mm , ff ) = paths * ( 1 ./ group_speed( : , mm , ff ) );
    end
    Mode_Save( ff ) = mm;
end
Data0 = Data2( ¬isnan( Data2 ) );

depth0 = depth;

inversion_setup
```

Preparation Algorithm

```
% Pass data to this algorithm
% 3D space and descretizations
% Source and Receiver Locations
% Frequencies, modes, travel times

% The data must be a column vector that groups frequency, then mode, then
% path

% paths_save = 1.000 * paths_save0;
depth_inacc = 0;
depth = depth0 + depth_inacc;

Max_Num_Iterations = 20;

DisCondepth = [];     % used by qualitative regularization; DisCondepth=[]
% means there are no discontinities in the profile
Weight_Option = 0;    % Decides if weighting is to occur - binary input

% Select points at which absolute constraint is applied
i = 0;
At_depth = 0;
for m = 1:length( y )
    for n = 1:length( x )
        i = i+1;

        % Only holds the top and bottom of the water column constant
        Absolute_Index_top( i ) = [ At_depth+1 ];
        Absolute_Index_bot( i ) = [ At_depth + depth(m,n)/dz+1 ];
        At_depth = At_depth + depth(m,n)/dz+1;

    end
end

% L and A are the same for all iterations (even though G might not be)
% because they are generated based on depth alone and only
% need to be determined once

% Relative Constraint (Tikhonov)
L = [];
for m = 1:round( yrange/dy )
```

```
    for n = 1:round( xrange/dx )

        [ Lt , ¬ ] = get_l( depth( m,n ) / dz + 1 , 2 );
        Lt = [ zeros( 1 , size( Lt , 2 ) ) ;
               Lt ;
               zeros( 1 , size( Lt , 2 ) ) ];
        L  = [ L , zeros( [ size( L , 1 ) , size( Lt , 2 ) ] ) ;
               zeros( [ size( Lt , 1 ) , size( L , 2 ) ] ) , Lt ];

    end

end

% Absolute Constraint
A_t = zeros( length( Absolute_Index_top ), sum( depth( : ) + 1 ) );
A_b = A_t;
for aa = 1:length( Absolute_Index_top )
    A_t( aa , Absolute_Index_top( aa ) ) = 1;
    A_b( aa , Absolute_Index_bot( aa ) ) = 1;
end

% Combination of constraints through lens of Tarantola's application of
% a priori data and model covariance
Cm0 = 5.5 ^ 2;
Cd0 = 100; % Amount to divide Data by to get a priori data variance

Cm  = eye( sum( depth( : ) + 1 ) ) / ( ( L.' * L ) * 5e2 + ...
    ( A_t.' * A_t ) * 1e-1 + ( A_b.' * A_b ) * 2e-1 ) + ...
    eye( sum( depth( : ) + 1 ) );
Cm  = Cm0 * Cm / mean( diag( Cm ) );


Data = Data0;

Background_Sediment_SSP = Csedt;

% water column sound speed profile of background environment
Background_H2O_SSP0 = NaN( max( depth( : ) ) , length( y ) * length( x ) );
% Assumes that a measurement was taken in one range block
% Fits a profile to that measurement and uses the fit as the
% basis for the background profile
i = 0;
wcn_fit = polyfit( dz * ( 0 : length( Sound_Speed_Measurement( ...
                  ¬isnan( Sound_Speed_Measurement ) ) )-1 ) , ...
                  Sound_Speed_Measurement( ...
```

```
                    ¬isnan( Sound_Speed_Measurement ) ) , 0 );

Background_H2O_SPP0_2 = nan( length( y ) , length( x ) , ...
                             max( depth( : ) ) + 1 );

for m = 1 : length( y )

    for n = 1 : length( x )

        i = i+1;
        Background_H2O_SSP( m , n , 1:depth( m,n )/dz+1 ) = ...
            polyval( wcn_fit , 0:dz:depth( m,n ) );

    end

end

Background_H2O_SSP0( Background_H2O_SSP0 == 0 ) = NaN;

minss = min( Background_H2O_SSP0( : ) );
maxss = max( Background_H2O_SSP0( : ) );
bckg  = 0;%-10:1.0:( ceil( maxss - minss ) +10 );

disp( 'Keyboard to check inputs.  No error has occurred.  F5 to continue' )
keyboard

% run the inverison algorithm
[ Rreg , inverted_h2o_ssp , inverted_sediment_ssp , resolution_matrix , ...
  last_iter , variance , G_final , p ] = ...
    inversion_algorithm( frequency , paths_save , Data , ...
                         Max_Num_Iterations , Background_Sediment_SSP , ...
                         Background_H2O_SSP , dx , xrange , dy , ...
                         yrange , dz , depth , Inversion_depth , Cm , ...
                         Cd0 , Mode_Save );

% Get the group speed for the inverted profile
[ Inverted_Group_Speed , ¬ , Inverted_Mode_Save , ¬ ] = ...
    Mod_Make_Background( frequency  , Mode_Save , x , y , dz , depth , ...
                         inverted_h2o_ssp( : , : , : , last_iter ) , ...
                         inverted_sediment_ssp , Inversion_depth , ...
                         Weight_Index );

% Generate the travel times for the inverted profiles
index = [ ];
Inverted_Data0 = nan( [ size( paths , 1 ) , max( Inverted_Mode_Save ) , ...
```

```
                    length( frequency ) ] );

for ff = 1 : length( frequency )

    for mm = 1 : size( Inverted_Group_Speed{ ff } , 3 )

        group_speed( : , mm , ff ) = ...
           reshape( permute( Inverted_Group_Speed{ ff }( : , : , mm ) , ...
                             [ 2 , 1 ] ) , length( x ) * length( y ) , 1 );
        Inverted_Data0( : , mm , ff ) = ...
            paths * ( 1 ./ group_speed( : , mm , ff ) );

    end

    index = [ index , ( 1:Inverted_Mode_Save( ff ) * ...
              size( paths , 1 ) ) + length( index ) ];

end
Inverted_Data = Inverted_Data0( ¬isnan( Inverted_Data0 ) );
% Take the difference between the true profile and the inverted profile
Δ_d = Data( index ) - Inverted_Data;

% Combine the residuals for each path into a single value
residue0 = mean( abs( Δ_d( Δ_d ≠ 0 )' ) , 2 );

% Save all the steps of the inversion and relabel
inverted_h2o_ssp_save = inverted_h2o_ssp; clear inverted_h2o_ssp
inverted_h2o_ssp = inverted_h2o_ssp_save(:,:,:,last_iter);

wssp_backgr   = Background_H2O_SSP;
wssp_inverted = inverted_h2o_ssp;

w_i_mean = mean( wssp_inverted,4 );

w_i_stdev = sqrt( variance( : , : , : , last_iter ) );

wssp_real_inverted = wssp_inverted( :,:,:,1 );

atdepth = 0;
model_covariance2 = diag( Cm );
model_covariance  = zeros( size( wssp2 ) );
for yy = 1:round( yrange / dy )
    for xx = 1:round( xrange / dx )

        model_covariance( yy , xx , 1:depth( yy , xx )+1 ) = ...
```

```
            model_covariance2( atdepth+1:atdepth+depth( yy , xx )+1 );
        atdepth = atdepth + depth( yy , xx )+1;

    end
end

wssp_backgr( wssp_backgr == 0 ) = NaN;

Profile_Visualization( w_i_mean , wssp_real_inverted , w_i_stdev , ...
                       wssp_backgr , wssp2 , [ xrange , yrange ] , dx , ...
                       dy , dz , depth , depth0 , S , R , wssp4 , ...
                       sqrt( model_covariance ) )
```

Inversion Algorithm

```
function [ residual , cWcn_backgrSave , cSed_backgrSave , R , iteration , ...
    variance , G , P ] = ...
    inversion_algorithm( freqs , paths , Data0 , num_iterations , ...
                         cSed_backgr , cWcn_backgr , dx , xrange , dy , ...
                         yrange , dz , wdepth , InversionDepth , Cm , ...
                         Cd0 , NumM )

dbstop if error

% freqs          = vector of frequencies used for the inversion [Hz].
% cgrp_true      = cell array of group speeds each cell contains a vector
%                  of wave numbers for the frequnecies specified by freqs
% Option         = what to invert for; (1) = sediment sound speed,(2) =
%                  water column sound speed, (3) = both water column and
%                  sediment sound speed
% num_iterations  = number of iterations to perform
% alpha1         = Lagrange multiplier - controls the weighting of the
%                  relative constraints
% alpha2         = Lagrange multiplier - controls the weighting of the
%                  absolute constraints
% cSed_backgr    = sediment sound speed profile for the background
%                  environment [meters/second].
% cWcn_backgr    = water column sound speed profile for the background
%                  environment [meters/second].
% dz             = depth descretization [meters]
% wdepth     = water depth [meters]
% InversionDepth = Depth for sediment inversion below the seafloor [meters]
%                  This does not depend on water depth.
% AbsoluteIndex  = indexs for the absolute constraints - perturbations from
%                  the mean will be minimized at these points. If you don't
%                  want to use absolute constraints, set AbsoluteIndex=[].
%                  and alpha2=0.
% WeightIndex    = cell array of indices of wave numbers to use in the
%                  inversion; each cell contains a vector of
%                  indices for the wave numbers specified by Ktrue.
If you
%                  want to use all the data, you may set WeightIndex=[].

P = 0;
variance  = zeros( round( yrange / dy ) , round( xrange / dx ) , ...
                   max( wdepth( : ) )+1 , num_iterations );
```

```
numfreq   = length( freqs );
iteration = 0;
s         = 0;
tau       = inf;

residual = NaN * ( 1:num_iterations );
cSed_backgrSave = NaN( [ size( cSed_backgr ) , num_iterations ] );
cWcn_backgrSave = NaN( [ size( cWcn_backgr ) , num_iterations ] );
while ( iteration < num_iterations ) && ( iteration < 5 * tau )

clear G residual_ffnum We DataWeights DataWeight d_b
iteration = iteration+1;

% Determine a residual decay rate with iterations assuming exp fit,
% this decay rate is used to establish convergence
% At least 3 iterations are required to estimate the decay rate

fprintf( 'The iteration number is %d.\n' , iteration );
clear d

cSed_backgrSave( :,:,iteration ) = cSed_backgr;
cWcn_backgrSave( :,:,:,iteration ) = cWcn_backgr;

% Background values for sound speed and density
c0  = ( cWcn_backgr );
rho = 1.0*ones( size( cWcn_backgr ) );
rhoc3   = rho .* c0 .^3;

index = [ ];
NMsave = linspace( 0 , 0 , numfreq );
for ffnum = 1:numfreq

    dif = .5; % Difference for derivative
    i = 0;
    clear group_speed_back

    modes = zeros( round( yrange/dy ) , round( xrange/dx ) );
    phi_backgroundmst = zeros( round( yrange/dy ) , ...
        round( xrange/dx ) , ...
        size( cWcn_backgr , 3 ) , ...
        NumM( ffnum ) );
    phi_backgroundpst = phi_backgroundmst;
    phi_backgroundst  = phi_backgroundmst;
    group_speed_back0 = zeros( round( yrange/dy ) , ...
        round( xrange/dx ) , NumM( ffnum ) );
```

```
    wave_num_back0    = group_speed_back0;

    % Get mode functions, group speeds, and wave numbers
for n = 1:round( yrange/dy )

    for m = 1:round( xrange/dx )

        i = i+1;

        % get info for freqs above and below freqs of interest
        % for derivative purposes
        [ cgrpm    ,  ¬  , phi_backgroundm ] = ...
            run_kraken( freqs( ffnum ) - dif , NumM( ffnum ) , ...
                        wdepth( n,m ) , dz , cWcn_backgr( n , m , ...
                        1:wdepth( n,m )/dz + 1 ) , cSed_backgr , ...
                        InversionDepth );

        [ cgrpp    ,  ¬  , phi_backgroundp ] = ...
            run_kraken( freqs( ffnum ) + dif , NumM( ffnum ) , ...
                        wdepth( n,m ) , dz , cWcn_backgr( n , m , ...
                        1:wdepth( n,m )/dz + 1 ) , cSed_backgr , ...
                        InversionDepth );

        [ cgr_bkgr , k_b , phi_background ]  = ...
            run_kraken( freqs( ffnum )       , NumM( ffnum ) , ...
                        wdepth( n,m ) , dz , cWcn_backgr( n , m , ...
                        1:wdepth( n,m )/dz + 1 ) , cSed_backgr , ...
                        InversionDepth );

        modes( n , m ) = min( [ length( cgrpm(    cgrpm    ≠ 0 ) ) , ...
                                length( cgrpp(    cgrpp    ≠ 0 ) ) , ...
                                length( cgr_bkgr( cgr_bkgr ≠ 0 ) ) , ...
                                NumM( ffnum ) ] );

        group_speed_back0( n , m , 1:modes( n , m ) , ffnum ) = ...
            cgr_bkgr( 1:modes( n , m ) );
        wave_num_back0(    n , m , 1:modes( n , m ) , ffnum ) = ...
            k_b( 1:modes( n , m ) );

        phi_backgroundmst( n , m , 1:wdepth( n,m )/dz + 1 , ...
            1:modes( n , m ) ) = ...
            phi_backgroundm( 1:wdepth( n,m )/dz + 1 , 1:modes( n , m ) );
        phi_backgroundpst( n , m , 1:wdepth( n,m )/dz + 1 , ...
            1:modes( n , m ) ) = ...
            phi_backgroundp( 1:wdepth( n,m )/dz + 1 , 1:modes( n , m ) );
```

```
            phi_backgroundst(  n , m , 1:wdepth( n,m )/dz + 1 , ...
                1:modes( n , m ) ) = ...
                phi_background(  1:wdepth( n,m )/dz + 1 , 1:modes( n , m ) );

        end

    end

        Paths = size( paths , 3 );

        % Determine maximum number of modes allowed based on ALL information
        NL  = min( [ size( phi_backgroundmst , 3 ) ;
                     size( phi_backgroundst , 3 ) ;
                     size( phi_backgroundpst , 3 ) ] );
        NM  = min( modes( : ) );

        cgr_back = group_speed_back0( : , : , 1:NM , ffnum  );
        k_b = wave_num_back0( : , : , 1:NM , ffnum );

        % Create travel times between all sources and receivers based present
        % profiles' group speeds
        paths_2d       = reshape( permute( paths , [ 2 , 1 , 3 ] ) , ...
                                  round( xrange/dx ) * round( yrange/dy ) , ...
                                  Paths ).';
        group_speed_2d = reshape( permute( cgr_back , [ 2 , 1 , 3 ] ) , ...
                                  round( xrange/dx ) * round( yrange/dy ) , ...
                                  NM );

        d_b = paths_2d * ( 1 ./ group_speed_2d );
        d_b = d_b( : );

        phi_backgroundmst = phi_backgroundmst( : , : , 1:NL , 1:NM );
        phi_backgroundpst = phi_backgroundpst( : , : , 1:NL , 1:NM );
        phi_backgroundst  = phi_backgroundst(  : , : , 1:NL , 1:NM );

        zm = abs( phi_backgroundst );

        % Derivative of phi
        zmd = ( abs( phi_backgroundpst ) .^2 - ...
                abs( phi_backgroundmst ) .^2 ) / ( 2*dif ) / ( 2*pi );

        % Determine G
        paths_5d    = repmat( permute( paths    , [ 1 , 2 , 4 , 3 , 5 ] ) , ...
                      [ 1 , 1 , size( rhoc3 , 3 ) , 1 , size( k_b , 3 ) ] );
        rhoc3_5d    = repmat( permute( rhoc3    , [ 1 , 2 , 3 , 4 , 5 ] ) , ...
```

```
              [ 1 , 1 , 1 , size( paths , 3 ) , size( k_b , 3 ) ] );
k_b_5d      = repmat( permute( k_b      , [ 1 , 2 , 5 , 4 , 3 ] ) , ...
              [ 1 , 1 , size( rhoc3 , 3 ) , size( paths , 3 ) , 1 ] );
zm_5d       = repmat( permute( zm       , [ 1 , 2 , 3 , 5 , 4 ] ) , ...
              [ 1 , 1 , 1 , size( paths , 3 ) , 1 ] );
zmd_5d      = repmat( permute( zmd      , [ 1 , 2 , 3 , 5 , 4 ] ) , ...
              [ 1 , 1 , 1 , size( paths , 3 ) , 1 ] );
cgr_back_5d = repmat( permute( cgr_back , [ 1 , 2 , 5 , 4 , 3 ] ) , ...
              [ 1 , 1 , size( rhoc3 , 3 )   , size( paths , 3 ) , 1 ] );

G_5d = ( paths_5d .* ( 2*pi*freqs( ffnum ) ) * dz ./ ...
       ( rhoc3_5d .* k_b_5d ) ) .* ( 2 * zm_5d .^ 2 + ...
       ( 2*pi*freqs( ffnum ) ) * zmd_5d - ( 2*pi*freqs( ffnum ) ) ./ ...
       ( k_b_5d .* cgr_back_5d ) .* zm_5d .^ 2 );

G_ffnum = reshape( ...
            permute( ...
                reshape( ...
                    permute( ...
                        G_5d , ...
                    [ 2 , 1 , 3 , 4 , 5 ] ) , ...
                round( xrange/dx ) * round( yrange/dy ) , ...
                max( wdepth( : ) )+1 , size( paths , 3 ) , NM ) , ...
            [ 3 , 4 , 2 , 1 ] ) , ...
          size( paths , 3 ) * NM , size( zm , 3 ) * ...
          size( paths , 1 ) * size( paths , 2 ) );

G_ffnum = G_ffnum( : , ¬isnan( sum( G_ffnum , 1 ) ) );

% An idea on how to remove rows with nans
% take sum so you have same number of rows but one column
% I = ¬isnan( summed );
% G = G( I , : );

clear co rho0 sR k_back group_speed_backack n
if ¬exist( 'G' , 'var' )
    G = G_ffnum;
    db0 = d_b;
else
    G = [ G ; G_ffnum ]; % stacks the information from each frequency
    db0 = [ db0 ; d_b ];
end

NMsave( ffnum ) = NM;
```

```
    % this next line is iffy
    index = [ index , ( 1:NM*Paths ) + ( sum( NMsave( : ) ) - ...
              NMsave( ffnum ) ) .* Paths ];

    clear wave_num_back0 phi_background phi_backgroundp phi_backgroundm ...
        G_ffnum d_ffnum d_b zmd zm phi_backgroundmst phi_backgroundpst ...
        phi_backgroundst

end

Data = Data0( index );

d = ( db0 - Data );

residual( iteration ) = sum( abs( d ) );
Cd = diag( ( Data / Cd0 ) .^ 2 );

% This is the inversion
q1  = ( G.' / Cd * G + eye( size( G , 2 ) ) / Cm ) \ G.' / Cd * d;
variance0 = diag( eye( size( G , 2 ) ) / ( G.' / Cd * G + ...
            eye( size( G , 2 ) ) / Cm ) );

if any( isnan( q1 ) ); keyboard; end

atdepth = 0;
qs = zeros( size( cWcn_backgr ) );
vs = qs;
for yy = 1:round( yrange / dy )
    for xx = 1:round( xrange / dx )

        qs( yy , xx , 1:wdepth( yy , xx )+1 ) = ...
            q1( atdepth+1:atdepth+wdepth( yy , xx )+1 );
        vs( yy , xx , 1:wdepth( yy , xx )+1 ) = ...
            variance0( atdepth+1:atdepth+wdepth( yy , xx )+1 );
        atdepth = atdepth + wdepth( yy , xx )+1;

    end
end

% This reorganizes the data from a vector into the appropriate matrix so it
% can be added to the current background profile
q = round( qs * 1e3 ) / 1e3;
variance( : , : , : , iteration ) = round( vs * 1e3 ) / 1e3;

R( : , : , iteration ) = G'/(G*G')*G;
```

```
    cWcn_backgr = cWcn_backgr+q;

r = d - G*q1;
s = sqrt(1/numel(r)*sum(r.^2)) + s;

cSed_backgrSave( :,:,iteration ) = cSed_backgr;
cWcn_backgrSave( :,:,:,iteration ) = cWcn_backgr;

if iteration > 5

    t   = 0:iteration-1;
    rel = residual( t+1 );
    p0  = [ mean( rel ) , ( max( rel ) - min( rel ) ) , iteration / 2 ];
    fh  = @( x , p ) p( 1 ) + p( 2 ) * exp( -x / p( 3 ) );
    errfh = @( p , x , y ) sum( ( y( : ) - fh( x( : ) , p ) ) .^ 2 );
    P   = fminsearch( errfh , p0 , [] , t , rel );
    tau = P( 3 );

end

end
```

Path Length Algorithm

```
function [ paths ] = Make_paths_better( all_x , all_y , ...
                     Source_Location , Receiver_Location )

% all_x is a vector of the x grid lines in the search area (including borders)
% all_y is a vector of the y grid lines in the search area (including borders)
% Source_Locations is a matrix with the coordinates of the sources
% Receiver_Locations is a matrix with the coordinates of the receivers

% paths returns a 3D matrix that is the size of the search environement in
% the first 2 dimensions and the number of sources times the number of
% receivers in the third dimension

S = Source_Location;    % Shorten variable for convenience
R = Receiver_Location;  % Shorten variable for convenience

NoS = size( S , 1 );        % Get the number of sources
NoR = size( R , 1 );        % Get the number of sources

% Generate an empty path matrix (must be zeros and not NaNs)
paths = zeros( length( all_y ) - 1 , length( all_x ) - 1 , NoS*NoR );

dx = mean( diff( all_x ) );     % Get x grid spacing (assumed uniform)
dy = mean( diff( all_y ) );     % Get y grid spacing (assumed uniform)

% It would not be terribly difficult to modify this code to allow for
% variable grid spacing, but that would make the inversion difficult...

% Loop through each path
count = 0;
for n = 1:NoS

    for m = 1:NoR
        count = count+1;

        % Determine the grid lines crossed by the path, logicals used for
        % differences in left-right and up-down relative location
        x = all_x( ( all_x > S( n , 1 ) & all_x < R( m , 1 ) ) ...
                 | ( all_x < S( n , 1 ) & all_x > R( m , 1 ) ) );
        y = all_y( ( all_y > S( n , 2 ) & all_y < R( m , 2 ) ) ...
                 | ( all_y < S( n , 2 ) & all_y > R( m , 2 ) ) );
```

```
% Determine equation of line to get corresponding point at grid crossings
% Turn off warnings because only 2 points are used for fitting
warning( 'off' , 'all' )
p_x = polyfit( [ S( n , 1 ) , R( m , 1 ) ] , ...
               [ S( n , 2 ) , R( m , 2 ) ] , 1 );
p_y = polyfit( [ S( n , 2 ) , R( m , 2 ) ] , ...
               [ S( n , 1 ) , R( m , 1 ) ] , 1 );
warning( 'on'  , 'all' )

% Evaluate corresponding points at grid crossings
y_x = polyval( p_x , x );
x_y = polyval( p_y , y );

% Sort grid crossings and source/receiver locations
[ x_tot , I ] = unique( [ S( n , 1 ) , x , x_y , R( m , 1 ) ] );
y_tot = [ S( n , 2 ) , y_x , y , R( m , 2 ) ];
y_tot = y_tot( I );

% The steps that use ?_tra are fixes corresponding to errors from
% numerical precision and are incredibly offensive
x_tra = mod( x_tot , dx );
y_tra = mod( y_tot , dy );

x_tra( x_tra < 1e-9 ) = 0;
y_tra( y_tra < 1e-9 ) = 0;

x_tot = floor( round( x_tot / dx * 1e9 ) / 1e9 ) * dx + x_tra;
y_tot = floor( round( y_tot / dy * 1e9 ) / 1e9 ) * dy + y_tra;

% Get the distance between each crossing
dist  = hypot( diff( x_tot ) , diff( y_tot ) );

% Determine which block each distance is travelled
x_low = x_tot( 1:end-1 );
y_low = y_tot( 1:end-1 );

x_hgh = x_tot( 2:end );
y_hgh = y_tot( 2:end );

x_block = ceil( max( x_hgh , x_low ) / dx );
y_block = ceil( max( y_hgh , y_low ) / dy );

% Update the paths matrix with the distance through each block
for ii = 1:length( dist )
```

```
            paths( y_block( ii ) , x_block( ii ) , count ) = ...
                dist( ii ) + paths( y_block( ii ) , x_block( ii ) , count );

        end

    end

end

% Round to throw away negligible distances
paths = round( paths * 1e6 ) / 1e6;









end
```

Bibliography

[1] Xavier Lurton and D. R. Jackson. *An Introduction to Underwater Acoustics*. Springer, New York, second edition, 2010.

[2] Subramaniam D. Rajan, James F. Lynch, and George V. Frisk. Perturbative inversion methods for obtaining bottom geoacoustic parameters in shallow water. *Journal of the Acoustical Society of America*, 82:998–1017, 1987.

[3] Walter E. Medeiros and Joao B. C. Silva. Geophysical inversion using approximate equality constraints. *Geophysics*, 61:1678–1688, 1996.

[4] Megan S. Ballard and Kyle M. Becker. Optimized constraints for the linearized geoacoustic inverse problem. *Journal of the Acoustical Society of America*, 129:652–661, 2011.

[5] Stan E. Dosso, Michael J. Wilmut, and Anna-Liesa S. Lapinski. An adaptive-hybrid algorithm for geoacoustic inversion. *IEEE Journal of Oceanic Engineering*, 26:324–336, 2001.

[6] N. Ross Chapman, S. Chin-Bing, D. King, and R. B. Evans. Benchmarking geoacoustic inversion methods for range-dependent waveguides. *IEEE Journal of Oceanic Engineering*, 28:320–330, 2003.

[7] Peter Gerstoft. Inversion of seismoacoustic data using genetic algorithms and a posteriori probability distributions. *Journal of the Acoustical Society of America*, 95:770–782, 1994.

[8] Timothy F. Duda, Richard A. Pawlowicz, James F. Lynch, and Bruce D. Cornuelle. Simulated tomographic reconstruction of ocean features using drifting acoustic receivers and a navigated source. *Journal of the Acoustical Society of America*, 96:2270–2279, 1995.

[9] B. Cornuelle, W. Munk, and P. Worcester. Ocean acoustic tomography from ships. *Journal of Geophysical Research*, 94:6232–6250, 1989.

[10] Fabienne Gaillard. Ocean acoustic tomography with moving sources or receivers. *Journal of Geophysical Research*, 90:11,891–11,898, 1985.

[11] Subramaniam D. Rajan and Kyle M. Becker. Inversion for range-dependent sediment compression-wave-speed profiles from modal dispersion data. *IEEE Journal of Oceanic Engineering*, 35:43–58, 2010.

[12] James F. Lynch, Subramaniam D. Rajan, and George V. Frisk. A comparison of broadband and narrow-band modal inversions for bottom geoacoustic proeprties at a site near Corpus Christi, Texas. *Journal of the Acoustical Society of America*, 89:648–665, 1990.

[13] Kazuhiko Ohta and George V. Frisk. Modal evolution and inversion for seabed geoacoustic properties in weakly range-dependent shallow-water waveguides. *IEEE Journal of Oceanic Engineering*, 22:501–521, 1997.

[14] George V. Frisk, James F. Lynch, and Subramaniam D. Rajan. Determination of compressional wave speed profiles using modal inverse techniques in a range-dependent environment in Nantucket Sound. *Journal of the Acoustical Society of America*, 86:1928–1939, 1989.

[15] Megan S. Ballard and Kyle M. Becker. Inversion for range-dependent water column sound speed profiles on the New Jersey shelf using a linearized perturbative method. *Journal of the Acoustical Society of America*, 127:3411–3421, 2010.

[16] Megan S. Ballard. *Optimized Constraints for the Linearized Geoacoustic Inverse Problem*. PhD thesis, The Pennsylvania State University, 2009.

[17] Finn B. Jensen, William A. Kuperman, Michael B. Porter, and Henrik Schmidt. *Computational Ocean Acoustics*. American Institute of Physics, New York, 1994.

[18] D. E. Weston. Intensity-range relations in oceanographic acoustics. *Journal of Sound and Vibration*, 18:271–287, 1971.

[19] Walter A. Strauss. *Partial Differential Equations: An Introduction*. John Wiley & Sons, Ltd, New York, Second edition, 2008.

[20] Philip M. Morse and K. Uno Ingard. *Theoretical Acoustics*. Princeton University Press, Princeton, New Jersey, First edition, 1968.

[21] E.G. Williams. *Fourier Acoustics: Sound Radiation and Nearfield Acoustical Holography*. Academic Press, Cambridge, 1999.

[22] James Ward Brown and Ruel V. Churchill. *Complex Variables and Applications*. McGraw-Hill Higher Education, New York, Eighth edition, 2009.

[23] David T. Blackstock. *Fundamentals of Physical Acoustics*. John Wiley & Sons, Inc., New York, 2000.

[24] Travis L. Poole. *Geoacoustic Inversion by Mode Amplitude Perturbation*. PhD thesis, The Massachusettes Institute of Technology, 2000.

[25] William Menke. *Geophysical Data Analysis: Discrete Inverse Theory Revised Edition*. Academic Press Limited, San Diego, 1989.

[26] Brian Borchers Richard C. Aster and Clifford H. Thurber. *Parameter Estimation and Inverse Problems*. Elsevier, 2005.

[27] Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. The Society for Industrial and Applied Mathematics, Philadelphia, 2005.

[28] George Backus and Freeman Gilbert. The resolving power of gross earth data. *Geophysical Journal of the Royal Astronomical Society*, 16:169–205, 1968.

[29] Timothy F. Duda, Ying-Tsong Lin, Arthur E. Newhall, Weifeng Gordon Zhang, and James F. Lynch. Computational studies of time-varying three-dimensional acoustic propagation in canyon and slope regions. *IEEE*, pages 1–6, 2010.

[30] Michael B. Porter. The KRAKEN normal mode program. `http://oalib.hlsresearch.com/Modes/AcousticsToolbox/manual_html/kraken.html`, 1997.

[31] Kenneth V. Mackenzie. Nine-term equation for sound speed in the oceans. *Journal of the Acoustical Society of America*, 70:807–812, 1981.